



**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN**

**CLIENTE MOODLE PARA ANDROID**

**Enrique Castillo Montaña**

**Cádiz, Enero 2014**





ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

CLIENTE MOODLE PARA ANDROID

DEPARTAMENTO: LENGUAJES Y SISTEMAS INFORMÁTICOS

DIRECTOR DEL PROYECTO: MANUEL PALOMO DUARTE

AUTOR DEL PROYECTO: ENRIQUE CASTILLO MONTAÑO

Cádiz, Enero 2014

Fdo: Enrique Castillo Montaña



*A mis padres y hermanos,  
su apoyo ha sido incondicional.*

*A mi tutor de proyecto,  
por su entrega y colaboración.*

*A Daniela,  
por aguantarme y animarme  
a llevar a cabo este PFC.*

Producto licenciado bajo GPL. Se incluyen los términos de la licencia al final del documento.

Copyright (c) 2013 Enrique Castillo Montaña.

# Índice General

<b>1. <a href="#">Introducción</a></b>	<b>1</b>
1.1. <a href="#">Objetivos</a>	1
1.2. <a href="#">Alcance</a>	1
1.3. <a href="#">Definiciones, acrónimos y abreviaturas</a>	1
1.4. <a href="#">Estructura del documento</a>	2
<b>2. <a href="#">Planificación</a></b>	<b>4</b>
<b>3. <a href="#">Descripción general</a></b>	<b>7</b>
3.1. <a href="#">Interfaces de usuario</a>	7
3.2. <a href="#">Interfaces software</a>	7
3.3. <a href="#">Ubicación</a>	8
<b>4. <a href="#">Introducción a Android</a></b>	<b>9</b>
4.1. <a href="#">Alternativas Moodle-Android</a>	13
<b>5. <a href="#">Análisis</a></b>	<b>15</b>
5.1. <a href="#">Metodología del desarrollo</a>	15
5.2. <a href="#">Requisitos del sistema</a>	15
5.2.1. <a href="#">Especificación de los requisitos funcionales</a>	15
5.2.2. <a href="#">Especificación de los requisitos de almacenamiento</a>	18
5.2.3. <a href="#">Especificación de los requisitos de rendimiento</a>	19
5.2.4. <a href="#">Especificación de los requisitos de idioma</a>	20
5.3. <a href="#">Análisis del sistema</a>	20
5.3.1. <a href="#">Modelos de casos de uso</a>	20
5.3.1.1. <a href="#">Casos de uso destinados a Login</a>	20
5.3.1.2. <a href="#">Casos de uso destinados a la subida de contenidos</a>	24
5.3.1.3. <a href="#">Casos de uso destinados a la gestión de participantes</a>	29
5.3.1.4. <a href="#">Casos de uso destinados a la gestión de contenidos</a>	35
5.3.1.5. <a href="#">Casos de uso destinados a los datos estadísticos</a>	43

5.3.2.	<a href="#">Modelo conceptual de datos</a>	44
5.3.2.1.	<a href="#">Diagrama de clases conceptuales</a>	44
<b>6.</b>	<b><a href="#">Diseño</a></b>	<b>46</b>
6.1.	<a href="#">Estructura de un proyecto Android</a>	47
6.2.	<a href="#">Diseño de la capa de gestión de datos</a>	50
6.2.1.	<a href="#">Diseño conceptual</a>	50
6.2.2.	<a href="#">Diseño lógico</a>	53
6.2.3.	<a href="#">Diseño físico</a>	54
6.3.	<a href="#">Diseño de la capa de dominio</a>	55
6.3.1.	<a href="#">Diagrama de clases de diseño</a>	55
6.3.2.	<a href="#">Diagrama de secuencia</a>	61
6.3.3.	<a href="#">Contratos de las operaciones</a>	69
6.4.	<a href="#">Diseño de la capa de presentación</a>	71
<b>7.</b>	<b><a href="#">Implementación</a></b>	<b>81</b>
7.1.	<a href="#">Instalación del entorno de trabajo</a>	81
7.1.1.	<a href="#">Requisitos del sistema</a>	81
7.1.2.	<a href="#">Instalación de Eclipse</a>	82
7.1.3.	<a href="#">Instalación del SDK Android</a>	82
7.1.4.	<a href="#">Instalación del plugin ADT de Eclipse</a>	83
7.2.	<a href="#">AndroidManifest.xml</a>	86
7.3.	<a href="#">Internacionalización</a>	89
7.4.	<a href="#">Problemas encontrados</a>	90
7.5.	<a href="#">Caché</a>	91
7.6.	<a href="#">Algoritmo AES</a>	93
<b>8.</b>	<b><a href="#">Pruebas</a></b>	<b>94</b>
8.1.	<a href="#">Genymotion</a>	94
8.2.	<a href="#">JUnit</a>	95
8.3.	<a href="#">UI Automator Viewer y uiautomator</a>	97
8.4.	<a href="#">Pruebas y validaciones</a>	99



8.4.1.	<a href="#">Plan de pruebas</a>	99
8.4.2.	<a href="#">Especificación del diseño de las pruebas</a>	100
8.4.3.	<a href="#">Especificación de los casos de pruebas</a>	101
8.4.4.	<a href="#">Especificación de los procedimientos de pruebas</a>	102
8.4.5.	<a href="#">Documentación de la ejecución de las pruebas</a>	102
<b>9.</b>	<b><a href="#">Conclusiones</a></b>	<b>103</b>
9.1.	<a href="#">Aspectos generales</a>	103
9.2.	<a href="#">Conocimientos adquiridos</a>	104
9.3.	<a href="#">Futuro del proyecto</a>	105
<b>10.</b>	<b><a href="#">Referencias y Bibliografía</a></b>	<b>106</b>
<b>A.</b>	<b><a href="#">Publicación de la aplicación</a></b>	<b>108</b>
<b>B.</b>	<b><a href="#">Manual de usuario</a></b>	<b>111</b>
<b>C.</b>	<b><a href="#">Manual de instalación</a></b>	<b>120</b>
<b>D.</b>	<b><a href="#">Ejemplos de la aplicación</a></b>	<b>123</b>
<b>E.</b>	<b><a href="#">Licencia</a></b>	<b>125</b>

# Índice de Figuras

Figura 2.1:	<a href="#">Diagrama de Gantt</a>	6
Figura 4.1:	<a href="#">Arquitectura de Android</a>	10
Figura 4.2:	<a href="#">Ciclo de vida de una aplicación Android</a>	12
Figura 4.3:	<a href="#">Iconos de las alternativas</a>	13
Figura 5.1:	<a href="#">Diagrama de casos de uso - Login</a>	20
Figura 5.2:	<a href="#">Diagrama de secuencia - Login</a>	21
Figura 5.3:	<a href="#">Diagrama de secuencia - Login cuenta almacenada</a>	22
Figura 5.4:	<a href="#">Diagrama de secuencia - Eliminar cuenta</a>	23
Figura 5.5:	<a href="#">Diagrama de casos de uso - Contenidos</a>	24
Figura 5.6:	<a href="#">Diagrama de secuencia - Subir foto</a>	25
Figura 5.7:	<a href="#">Diagrama de secuencia - Hacer foto y subirla</a>	26
Figura 5.8:	<a href="#">Diagrama de secuencia - Grabar vídeo y subirlo</a>	27
Figura 5.9:	<a href="#">Diagrama de secuencia - Grabar audio y subirlo</a>	28
Figura 5.10:	<a href="#">Diagrama de casos de uso - Participantes</a>	29
Figura 5.11:	<a href="#">Diagrama de secuencia - Listar participantes</a>	30
Figura 5.12:	<a href="#">Diagrama de secuencia - Detalles participante</a>	31
Figura 5.13:	<a href="#">Diagrama de secuencia - Enviar mensaje</a>	32
Figura 5.14:	<a href="#">Diagrama de secuencia - Añadir nota</a>	33
Figura 5.15:	<a href="#">Diagrama de secuencia - Agregar a la agenda</a>	34
Figura 5.16:	<a href="#">Diagrama de casos de uso - Cursos</a>	35
Figura 5.17:	<a href="#">Diagrama de secuencia - Listar cursos</a>	36
Figura 5.18:	<a href="#">Diagrama de secuencia - Acceder curso</a>	37
Figura 5.19:	<a href="#">Diagrama de secuencia - Descargar fichero</a>	38
Figura 5.20:	<a href="#">Diagrama de secuencia - Visualizar fichero</a>	39
Figura 5.21:	<a href="#">Diagrama de secuencia - Acceder Url</a>	40
Figura 5.22:	<a href="#">Diagrama de secuencia - Acceder carpeta</a>	41
Figura 5.23:	<a href="#">Diagrama de secuencia - Versión web</a>	42
Figura 5.24:	<a href="#">Diagrama de secuencia - Ayuda</a>	42
Figura 5.25:	<a href="#">Diagrama de casos de uso - Estadísticas</a>	43

Figura 5.26:	<a href="#">Diagrama de secuencia - Dibujar estadísticas</a>	44
Figura 5.27:	<a href="#">Diagrama conceptual de clases</a>	45
Figura 6.1:	<a href="#">Arquitectura de nuestra aplicación</a>	47
Figura 6.2:	<a href="#">Esquema Entidad-Relación</a>	52
Figura 6.3:	<a href="#">Esquema Entidad-Relación - Conexiones al sistema</a>	52
Figura 6.4:	<a href="#">Secuencias para la creación de la base de datos</a>	55
Figura 6.5:	<a href="#">Diagrama de clases de diseño - Login</a>	56
Figura 6.6:	<a href="#">Diagrama de clases de diseño - Subida Fotos</a>	56
Figura 6.7:	<a href="#">Diagrama de clases de diseño - Subida Audios</a>	57
Figura 6.8:	<a href="#">Diagrama de clases de diseño - Cursos</a>	57
Figura 6.9:	<a href="#">Diagrama de clases de diseño - Participantes</a>	58
Figura 6.10:	<a href="#">Diagrama de clases de diseño - Estadísticas, ayuda y acceso Web</a>	58
Figura 6.11:	<a href="#">Diagrama de clases de diseño - Acceso a datos</a>	59
Figura 6.12:	<a href="#">Diagrama de clases de diseño - DTOs (parte I)</a>	59
Figura 6.13:	<a href="#">Diagrama de clases de diseño - DTOs (parte II)</a>	60
Figura 6.14:	<a href="#">Diagrama de clases de diseño - Servicios</a>	60
Figura 6.15:	<a href="#">Diagrama de clases de diseño - Utils</a>	61
Figura 6.16:	<a href="#">Diagrama de interacción - Login</a>	62
Figura 6.17:	<a href="#">Diagrama de interacción - Login desde cuenta almacenada</a>	62
Figura 6.18:	<a href="#">Diagrama de interacción - Principal</a>	63
Figura 6.19:	<a href="#">Diagrama de interacción - Subir foto almacenada</a>	63
Figura 6.20:	<a href="#">Diagrama de interacción - Realizar foto y subirla</a>	64
Figura 6.21:	<a href="#">Diagrama de interacción - Grabar audio y subirlo</a>	64
Figura 6.22:	<a href="#">Diagrama de interacción - Listar participantes</a>	65
Figura 6.23:	<a href="#">Diagrama de interacción - Detalle de un participante</a>	65
Figura 6.24:	<a href="#">Diagrama de interacción - Mandar mensaje a un participante</a>	65
Figura 6.25:	<a href="#">Diagrama de interacción - Cursos asignados al usuario</a>	66
Figura 6.26:	<a href="#">Diagrama de interacción - Detalle de un curso</a>	66
Figura 6.27:	<a href="#">Diagrama de interacción - Descargar contenido</a>	67
Figura 6.28:	<a href="#">Diagrama de interacción - Visualizar contenido carpeta</a>	67
Figura 6.29:	<a href="#">Diagrama de interacción - Estadísticas</a>	67
Figura 6.30:	<a href="#">Diagrama de interacción - Versión web</a>	68
Figura 6.31:	<a href="#">Diagrama de interacción - Ayuda Moodle</a>	68
Figura 6.32:	<a href="#">Diseño - View y Viewgroup</a>	71
Figura 6.33:	<a href="#">Diseño - Tipos de Views</a>	72

<b>Figura 6.34:</b>	<a href="#">Diseño de pantalla - Login</a>	73
<b>Figura 6.35:</b>	<a href="#">Diseño de pantalla - Login desde cuenta almacenada</a>	74
<b>Figura 6.36:</b>	<a href="#">Diseño de pantalla - Menú Principal</a>	74
<b>Figura 6.37:</b>	<a href="#">Diseño de pantalla - Pantalla Subidas</a>	75
<b>Figura 6.38:</b>	<a href="#">Diseño de pantalla - Listado de participantes</a>	76
<b>Figura 6.39:</b>	<a href="#">Diseño de pantalla - Detalle participante</a>	76
<b>Figura 6.40:</b>	<a href="#">Diseño de pantalla - Lista de cursos</a>	77
<b>Figura 6.41:</b>	<a href="#">Diseño de pantalla - Detalle curso</a>	78
<b>Figura 6.42:</b>	<a href="#">Diseño de pantalla - Elementos de una carpeta</a>	78
<b>Figura 6.43:</b>	<a href="#">Diseño de pantalla - Estadísticas</a>	79
<b>Figura 6.44:</b>	<a href="#">Diseño de pantalla - Versión Web</a>	80
<b>Figura 6.45:</b>	<a href="#">Diseño de pantalla - Ayuda</a>	80
<b>Figura 7.1:</b>	<a href="#">Descarga de Eclipse</a>	82
<b>Figura 7.2:</b>	<a href="#">Descarga de SDK Android</a>	83
<b>Figura 7.3:</b>	<a href="#">Instalación de Android Developer Tools</a>	84
<b>Figura 7.4:</b>	<a href="#">Especificar ubicación del SDK</a>	85
<b>Figura 7.5:</b>	<a href="#">ADV Manager</a>	86
<b>Figura 8.1:</b>	<a href="#">Genymotion</a>	95
<b>Figura 8.2:</b>	<a href="#">UI Automator Viewer</a>	98
<b>Figura B.1:</b>	<a href="#">Pantalla de Login</a>	112
<b>Figura B.2:</b>	<a href="#">Pantalla Principal</a>	113
<b>Figura B.3:</b>	<a href="#">Pantalla Subidas</a>	114
<b>Figura B.4:</b>	<a href="#">Pantalla Información Participante</a>	115
<b>Figura B.5:</b>	<a href="#">Pantalla Curso</a>	116
<b>Figura B.6:</b>	<a href="#">Pantalla Estadísticas</a>	117
<b>Figura B.7:</b>	<a href="#">Pantalla versión Web</a>	118
<b>Figura B.8:</b>	<a href="#">Pantalla Ayuda</a>	119
<b>Figura C.1:</b>	<a href="#">Configuración para instalar apk</a>	121
<b>Figura C.2:</b>	<a href="#">Configuración del sitio Moodle</a>	122

# ***Índice de Tablas***

Tabla 4.1:	<a href="#">Tabla comparativa de aplicaciones</a>	14
Tabla 6.1:	<a href="#">Tabla de entidades</a>	50
Tabla 6.2:	<a href="#">Tabla de atributos de la entidad Cuentas</a>	51
Tabla 6.3:	<a href="#">Tabla de atributos de la entidad Actividad</a>	52
Tabla 7.1:	<a href="#">Tabla de atributos AndroidManifest.xml</a>	88
Tabla 7.2:	<a href="#">Tabla de ejemplo de definición de una cadena</a>	89
Tabla 7.3:	<a href="#">Tabla de ejemplo de referencia a cadena desde 'layout'</a>	89
Tabla 7.4:	<a href="#">Tabla de ejemplo de uso de cadena desde el código</a>	90
Tabla 7.5:	<a href="#">Tabla Cache – variables timestamps</a>	92
Tabla 7.6:	<a href="#">Tabla Cache – comprobación timestamps</a>	92
Tabla 7.7:	<a href="#">Tabla AES – encriptación contraseña</a>	93
Tabla 8.1:	<a href="#">Tabla Junit – primer test</a>	96
Tabla 8.2:	<a href="#">Tabla Junit – segundo test</a>	97
Tabla 8.3:	<a href="#">Tabla uiautomator</a>	99

---

## ***Introducción***

---

La realización de este Proyecto Fin de Carrera tiene como objetivos generales el trabajo en 3 líneas: aplicar los conocimientos adquiridos en la titulación de *Ingeniería Técnica en Informática de Gestión*, realizar una pequeña contribución al *Software Libre* y, por último, el estudio y uso de nuevas herramientas para ampliar conocimientos sobre Informática.

### **1.1 Objetivos**

Este Proyecto consiste en la creación de una aplicación que nos permite acceder a sitios Moodle desde un terminal Android. No está orientado a un solo sitio Moodle, el cliente es genérico.

Podemos definir Moodle como una aplicación web para la creación de cursos virtuales basados en Internet y sitios web. Tiene fines educativos. En cuanto a Android, se trata de un sistema operativo de carácter abierto bastante extendido en el mundo de la telefonía móvil.

Como ya se ha indicado anteriormente, nuestra aplicación permitirá acceder a un sitio Moodle de forma directa desde un dispositivo con sistema operativo Android.

### **1.2 Alcance**

La aplicación será capaz de acceder a cualquier sitio Moodle de versión 2 o superior y que tenga habilitados los 'Mobile Web Services'.

### **1.3 Definiciones, acrónimos y abreviaturas**

Se listan a continuación los acrónimos y abreviaturas que irán apareciendo a lo largo del documento, con su correspondiente definición.

- **API:** conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- **GUI:** del inglés (graphical user interface), es el medio en el que el usuario se comunica con la máquina.
- **IDE:** entorno integrado de desarrollo. Aplicación compuesta por un conjunto de herramientas útiles para un programador.
- **SGBD:** Sistema de Gestión de Base de Datos. Conjunto de programas encargado de definir, construir y manipular una base de datos. Sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
- **SQLite:** sistema de gestión de base de datos relacional, multihilo y multiusuario. Usado en esta aplicación.
- **SQL:** lenguaje de consulta estructurado. Permite realizar diversas operaciones sobre la base de datos.
- **SVN:** subversion, un sistema de revisiones de versiones software.
- **SDK:** kit de desarrollo de software. Conjunto de herramientas de desarrollo de software.
- **ADT:** plugin Eclipse que contiene herramientas para el desarrollo de aplicaciones Android.
- **1FN:** primera forma normal. Usado en base de datos.
- **2FN:** segunda forma normal. Usado en base de datos.
- **3FN:** tercera forma normal. Usado en base de datos.
- **FNBC:** Forma Normal de Boyce-Codd. Usado en base de datos.

## 1.4 Estructura del documento

Para el desarrollo de esta memoria, se han seguido las pautas marcadas en el documento “Recomendaciones para la realización de la documentación del Proyecto Fin de Carrera”.

La presente memoria se divide en las siguientes partes:

- **Introducción:** incluye una breve descripción acerca del proyecto, se enumeran sus objetivos y su alcance. Por último, se detalla la estructura de este documento.
- **Planificación:** a través de un diagrama de “Gantt”, se detalla la planificación del proyecto y el porcentaje de esfuerzo dedicado a cada una de sus fases.
- **Descripción general:** visión en profundidad del proyecto. Se especifican las distintas tecnologías utilizadas para el desarrollo del mismo.

- **Introducción a Android:** se explica brevemente en qué consiste el sistema operativo Android y sus aplicaciones.
- **Análisis:** se definen los requisitos funcionales del sistema, diagramas de casos de uso, diagramas de secuencia y contrato de las operaciones.
- **Diseño:** diseño de la base de datos, y de los diagramas de secuencia y de clase aplicados al diseño.
- **Implementación:** aspectos más relevantes relacionados con la implementación de la aplicación. Además, se enumerarán los problemas encontrados en esta fase.
- **Pruebas:** se verifica que el software cumple su cometido.
- **Conclusiones:** valoración global, posibles mejoras y ampliaciones del proyecto.
- **Bibliografía y Referencias:** libros y referencias consultados para la elaboración del proyecto.
- **Apéndices:**
  - **Publicación de la aplicación:** opciones para difundir nuestro software.
  - **Manual de usuario:** manual de funcionamiento de la aplicación destinado a los usuarios finales.
  - **Manual de instalación:** instrucciones para una correcta instalación.
  - **Ejemplos de la aplicación:** se recogen un par de ejemplos prácticos de uso de la aplicación.
  - **Licencia:** texto de la licencia sobre la cual se rige el proyecto.



## Capítulo 2

---

### ***Planificación***

---

Durante el desarrollo del proyecto se ha seguido la planificación temporal que se detalla a continuación. Dicha planificación se verá reflejada en su correspondiente diagrama de Gantt.

El resumen de las tareas reflejadas en el gráfico es el siguiente:

- **Fase Inicial:** [03/06/2013 - 17/07/2013] El objetivo de esta fase es la puesta en marcha del proyecto.
  - Idea del proyecto: [03/06/2013 - 07/06/2013] Durante esta fase se define exactamente en que iba a consistir exactamente el proyecto
  - Entrevista con el tutor: [07/06/2013 - 18/06/2013] En este periodo se conciertan varias entrevistas con el tutor para exponerle la idea de proyecto. Finalmente se da vía libre a la realización del mismo.
  - Búsqueda de bibliografía: [18/06/2013 - 28/06/2013] Se recopila gran cantidad de información que puede ayudar al desarrollo del proyecto. Para ello, se utilizan libros de la biblioteca, contenidos de internet y apuntes de la titulación.
  - Creación del prototipo [28/06/2013 - 12/07/2013] Con ayuda de un editor gráfico, se crea un prototipo del aspecto que tendrá la interfaz gráfica del usuario.
  - Planificación [12/07/2013 - 17/07/2013] Como su propio nombre indica, en esta fase se planifican las tareas que se llevarán a cabo en la fase de ejecución.
- **Fase de Ejecución:** [19/07/2013 - 31/01/2014] Desarrollo del proyecto.
  - Análisis de requisitos: [19/07/2013 - 15/08/2013] Con toda la información recopilada anteriormente, se obtienen todos los requisitos de nuestra aplicación. El análisis debe ser exhaustivo ya que podemos encontrarnos con requisitos mal definidos o incompletos.

- Especificación: [13/08/2013 - 13/09/2013] Se define el comportamiento que tendrá la aplicación una vez se encuentre desarrollada. Dedicaremos el tiempo que sea necesario para que la aplicación cumpla toda su funcionalidad.
  - Diseño: [13/09/2013 - 25/10/2013] Es una de las etapas más importantes. Debemos ser muy meticulosos con el diseño de nuestra aplicación ya que ellos nos evitará retrasos en la implementación para la corrección de errores.
  - Codificación: [25/10/2013 - 31/12/2013] Se trata de la fase más larga. Se implementa todo el diseño de la fase anterior. Llegados a este punto, hay que decidir el IDE sobre el que vamos a trabajar, con qué lenguaje, gestor de la base de datos...
  - Pruebas: [31/12/2013 - 31/01/2014] Se comprueba que la aplicación cumple con lo indicado en la fase de especificación.
- 
- **Fase de Documentación:** [19/07/2013 - 04/02/2014] Al mismo tiempo que la fase de ejecución, se irá generando la documentación necesaria que finalmente quedará plasmada en esta memoria.
  - **Fase de Finalización:** [31/01/2013 - 14/04/2014] En esta fase se demuestra que nuestra aplicación es un producto de calidad y que cumple con los requisitos marcados.
    - Revisión del tutor: [31/01/2014 - 05/02/2014] El tuto asignado a nuestro proyecto debe comprobar si el trabajo cumple las condiciones e ideas iniciales.
    - Preparación del material: [07/02/2014 - 18/02/2014] Es el momento de revisar que la memoria es correcta, para mandarla a imprimir y encuadernar. Se procede a entregar el material a la Universidad y a cumplimentar la matrícula para la presentación del PFC.
    - Presentación: [14/04/2014] Exposición del trabajo que será evaluado por un Tribunal.

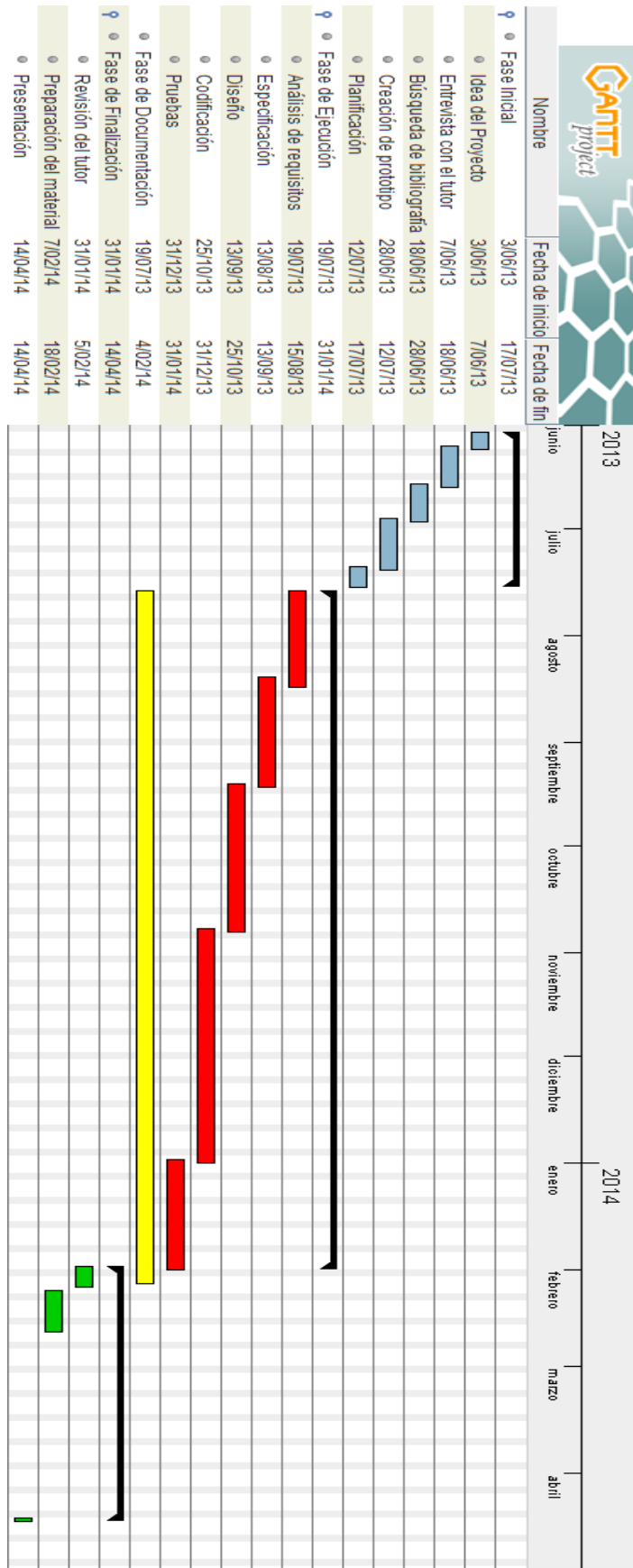


Figura 2.1: Diagrama de Gantt

---

### ***Descripción General***

---

Se trata de un proyecto de Software Libre, por tanto, cualquier persona podrá continuarlo en el futuro, ya sea el propio autor o personas ajenas al mismo.

Es un proyecto independiente; no se trata de una parte de otro ni la continuación de uno ya empezado.

#### ***3.1 Interfaces de usuario***

La interfaz de usuario se basa en una pantalla principal o de inicio a través de la cual, el usuario puede acceder a todas las funcionalidades que ofrece la aplicación. Para alcanzar dicha pantalla, antes ha tenido que superar un proceso de login al sitio deseado. Para ello, se encontrará una pantalla en la que introducir sus credenciales o acceder con datos ya almacenados.

Al tratarse de una aplicación dirigida a terminales móviles, el usuario no podrá modificar el tamaño de las pantallas. Siempre trabajará con una pantalla que ocupa la totalidad de la pantalla del terminal y desde ahí podrá ir accediendo a las distintas funcionalidades que se le ofrecen.

El idioma en el que aparecerá la aplicación dependerá del idioma en el que se encuentre el terminal. Con esto quiero decir que la aplicación desarrollada permite su internacionalización y soporta diferentes idiomas como español, inglés o francés.

#### ***3.2 Interfaces software***

La aplicación está preparada para funcionar sobre sistema operativo Android. Para el desarrollo de la misma, se han utilizados las Android SDK, que son un conjunto de librerías y utilidades proporcionadas por Google para el desarrollo de aplicaciones Android. La versión de

la SDK utilizada es la 8, con lo cual nuestra aplicación estará soportado en sistemas Android 2.2 o superior (más del 99% de cuota de mercado actual).

Ya que nuestra aplicación no utiliza características del SO avanzadas introducidas en las últimas versiones de Android, se ha optado por hacer la aplicación compatible con la versión más baja posible, con el objeto de llegar a un mayor número de teléfonos. La versión 8 de la API con la que es compatible la aplicación engloba a prácticamente la totalidad de terminales Android del mercado.

El lenguaje de programación utilizado ha sido Java. A través de la herramienta Eclipse SDK, que es un entorno de desarrollo Java de código abierto multiplataforma, se ha desarrollado la totalidad de la aplicación. Además, se ha utilizado el lenguaje XML para crear las vistas o interfaces con las que interactuará el usuario de la aplicación. Eclipse ofrece la posibilidad de crear esas vistas escribiendo directamente el código o a través de un visualizador interactivo al que iremos arrastrando los diferentes componentes de la interfaz mediante el método de *'drag and drop'* (pica y arrastra).

El SDK de Android provee funcionalidades para ayudar al programador a implementar una base de datos propia para su aplicación. Las BBDD Android se basan en el sistema SQLite3, que es un sistema de gestión de base de datos ligero que administra el propio terminal.

### **3.3 Ubicación**

Para poder hacer uso de la aplicación, tendremos que disponer de acceso a Internet en nuestro terminal, ya que, de lo contrario, nos será imposible conectarnos a los diferentes sitios Moodle.

## Capítulo 4

---

### ***Introducción a Android***

---

Según se define en Wikipedia, “*Android es un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005.*<sup>10</sup>*Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles.*<sup>11</sup>*El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008”.*

Concretamente, está basado en un núcleo Linux 2.6 cuya función es hacer de capa de abstracción entre el hardware y el resto del sistema, ofreciendo a la capa del software los diversos servicios existentes. Entre estos servicios se encuentran la seguridad, la gestión de procesos y memoria, la pila de red y el modelo de drivers. Ya por encima del núcleo se han diseñado una serie de capas que completan un entorno de desarrollo asequible para cualquier programador. Se define el ‘runtime’ como el conjunto de bibliotecas que ofrecen la mayoría de las funcionalidades disponibles en el núcleo de Java, lenguaje mediante el cual está programado el sistema.

Uno de sus principales componentes es la Máquina Virtual Dalvik. Permite que un dispositivo pueda ejecutar varias máquinas virtuales, de modo que cada aplicación sea lanzada como un proceso independiente y con su propia instancia de la máquina virtual, a la vez que supone un bajo consumo de recursos.

El conjunto de bibliotecas, escritas en C/C++, son usadas en muchos programas Android, y están puestas a disposición del programador mediante el Framework de aplicaciones. Sus competencias abarcan desde la propia gestión del sistema, como System C, hasta la manejo de bases de datos SQLite, pasando por diversas librerías multimedia con soporte para audio y video 2D y 3D, como Surface Manager, Media Framework, OpenGL, o FreeType, o la navegación web facilitada por el motor LibWebCore, entre otras funcionalidades.

El Framework de Android es el componente que da soporte a los desarrolladores para el uso de las características anteriormente expuestas, y es, además, el mismo sobre el que están

diseñadas las aplicaciones propias incluidas en la plataforma Android. Permite una sencilla reutilización de componentes y comunicación entre aplicaciones, siempre sujetas a ciertas medidas de seguridad que facilitan, por ejemplo, la actualización o sustitución de componentes por parte del usuario, resultando un sencillo y efectivo método para utilizar novedades o introducir mejoras en el software. En la figura 4.1, obtenida de <http://www.somoslibres.org>, podemos ver un esquema con la arquitectura de Android.

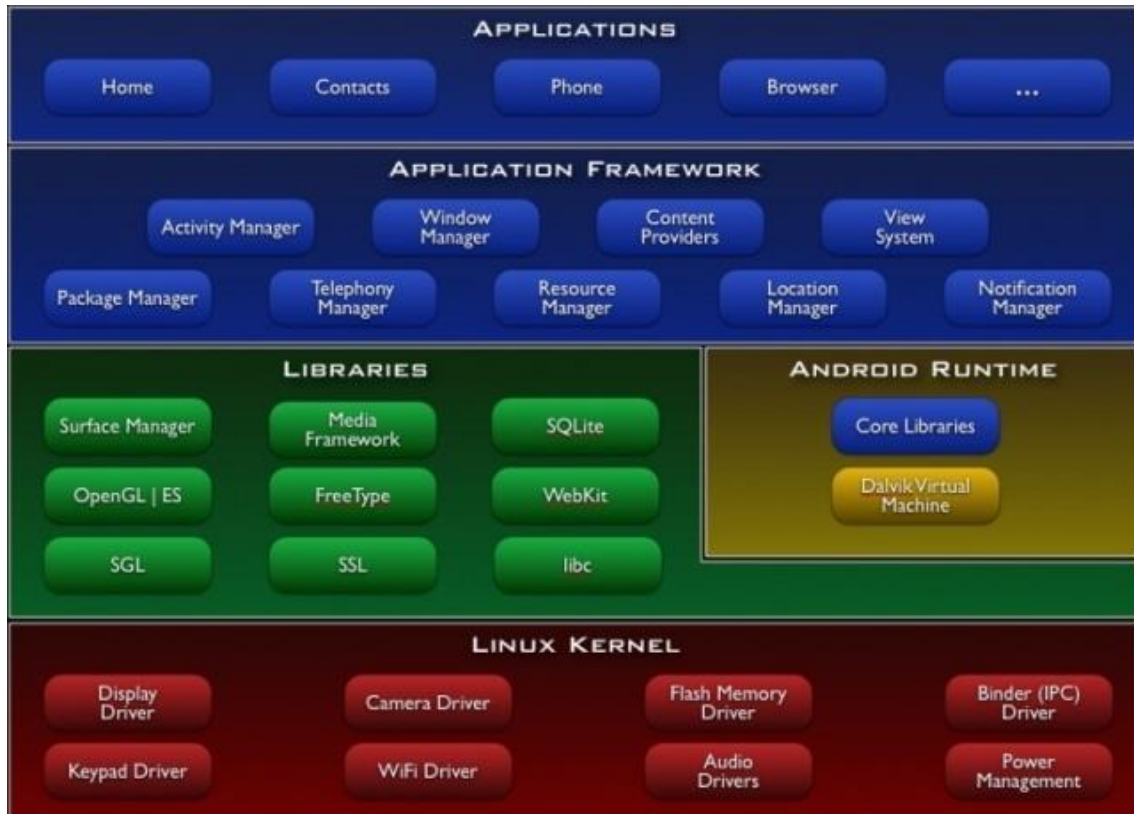


Figura 4.1: Arquitectura de Android

Los principales conjuntos de servicios ofrecidos son los siguientes:

- Un extenso y variado conjunto de vistas (Views) ofrecidas para el diseño de interfaces gráficas de usuario y su interactividad con el sistema, como los típicos botones, cuadros de texto o listas.
- Los proveedores de contenidos (Contents Providers) son el método de intercambio de información entre aplicaciones, ya sea compartiendo los datos propios o accediendo a los de otras aplicaciones.
- Los gestores de recursos (Resources Manager) permiten el acceso indexado a recursos como cadenas o gráficos, en un intento de modular aún más el diseño de las aplicaciones y el uso de sus recursos.

- El gestor de notificaciones (Notification Manager) dirige alertas personalizadas al software, que son mostradas en una barra de estado.
- Finalmente, el gestor de actividades (Activities Manager) es responsable del ciclo de vida de las aplicaciones.

Una vez proporcionados todos estos estratos, Android proporciona una serie de aplicaciones base, programadas en Java, que facilitan el manejo de las características del sistema, tales como un cliente de correo electrónico y SMS, calendario, mapas, navegador o gestor de contactos, entre otros. Sobre esta misma capa sería donde se sustenten las aplicaciones de usuario diseñadas por terceros desarrolladores.

El resto de características del sistema Android podríamos agruparlas por conectividad (telefonía GSM, Bluetooth, EDGE, WIFI, 3G), formatos multimedia soportados (MPEG4, H.264, MP3, OGG, AAC, AMR, JPG, PNG, GIF) y otras características hardware, tales como GPS, brújula, acelerómetro, cámara o pantalla táctil.

En el ámbito de desarrollo, dispone de un complemento para el IDE Eclipse, así como herramientas de simulación y depuración. Las aplicaciones creadas pueden ponerse a disposición del usuario a través de Google Play, donde podrán ser descargadas e instaladas por los usuarios, sirviendo igualmente como fuente de reporte de fallos y errores en el funcionamiento.

Otra peculiaridad de Android es el ciclo de vida de las aplicaciones, ya que no se trata únicamente de abrir y cerrar a gusto del usuario, sino que estas, una vez iniciadas, permanecen cargadas en memoria siempre que se disponga de recursos para ello. En caso contrario el propio sistema operativo se encargaría de destruirlas definitivamente. Dicho ciclo de vida se rige por las llamadas a los métodos onCreate, onStart, onResume, onPause, onStop, onStop, onDestroy y onRestart. En la siguiente página se muestra un gráfico sacado de <http://droideando.blogspot.com.es> en el que aparece reflejado de una manera más intuitiva el ciclo anteriormente descrito.



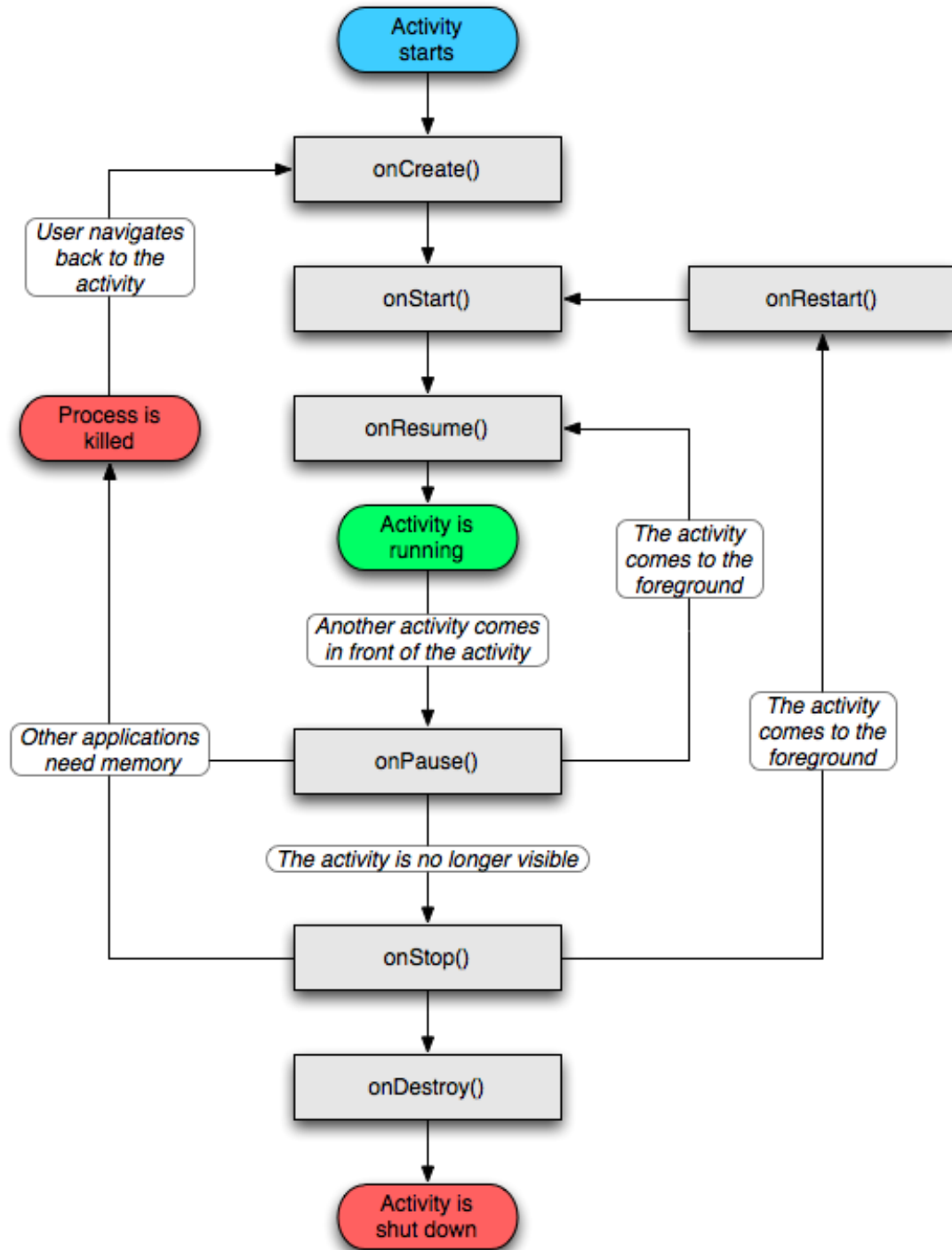


Figura 4.2: Ciclo de vida de una aplicación Android

## 4.1 Alternativas Moodle-Android

Haciendo una búsqueda de clientes para Moodle en el mercado oficial de aplicaciones de Android (Google Play), los resultados son escasos. Ninguna de las aplicaciones ofrece las funcionalidades que presenta la aplicación desarrollada en este Proyecto. La mayoría están muy mal valoradas por la comunidad de usuarios, otras no son un cliente genérico que se pueda conectar a cualquier servidor moodle mientras que el resto no muestran una interfaz práctica y amigable para el usuario.

A continuación, se listan los 3 resultados más relevantes en una búsqueda de 'Moodle' en el mercado de aplicaciones Google Play:



Figura 4.3: Iconos de las alternativas

- MDroid

<https://play.google.com/store/apps/details?id=in.co.praveenkumar>

Es una aplicación que aún cuenta con pocas descargas y parece encontrarse en un estado de madurez bajo. Al intentar conectar la aplicación con nuestro sitio Moodle de pruebas, emite un error y se cierra inesperadamente.

- Droodle

<https://play.google.com/store/apps/details?id=com.void.droodle>

La aplicación tiene una interfaz muy poco usable y la mayoría de sus valoraciones son negativas. El testeo de la aplicación nos demuestra que, tras hacer login correctamente, el sistema se limita a mostrar los cursos y no ofrece la posibilidad de ver los contenidos, ni los participantes ni de subir ningún tipo de fichero.

- UC3Moid

<https://play.google.com/store/apps/details?id=uni.net.moodle>

Es la aplicación mejor valorada de las presentes en Google Play. Sin embargo, se limita a dar acceso a los cursos y contenidos de una clase concreta de la Universidad Carlos III de Madrid.

Comentar también la problemática de navegar por un sitio web de Moodle desde el propio navegador del terminal. Este procedimiento sería lento y complicado, por ello resulta bastante práctico el uso del cliente implementado en este trabajo.

Por último, indicar que nuestro cliente se asemeja bastante al ya existe para terminales con sistema operativo iOS. Nos hemos basado en esta aplicación debido a sus buenas valoraciones y a que explota de la manera más clara posible todos los recursos ofrecidos a través de los Mobile Web Services de Moodle. El enlace para la descarga de dicha aplicación se encuentra en:

<https://itunes.apple.com/es/app/moodle-mobile/id633359593?mt=8>

				
Interfaz amigable	Si	No	No	No
Universal	Si	No	Si	No
Multiidioma	Si	No	No	No
Explota todos los 'Mobile Web Services'	Si	Si	No	Si
Estadísticas	Si	No	No	No

Tabla 4.1: Comparativa de aplicaciones

---

# **Análisis**

---

En este apartado se detallarán las diferentes fases del análisis previo realizado antes de abordar la implementación del cliente Moodle. Primero, se hará una definición detallada de los requisitos que ha de cumplir la aplicación, tanto los referentes a la funcionalidad como al conjunto de datos que ésta debe almacenar. Adicionalmente, se detallarán también algunos requisitos de rendimiento que se han identificado al comprobar que la implementación básica hacía demasiadas peticiones al servidor de Moodle y, por tanto, tenía un bajo rendimiento y sobrecargaba innecesariamente el tráfico de red. También se incluyen los requisitos correspondientes a los datos estadísticos que estarán disponibles para la consulta del usuario.

### **5.1 Metodología del desarrollo**

Para el desarrollo del proyecto se ha utilizado una metodología basada en el “Proceso Unificado Racional” (Rational Unified Process (RUP)). Este proceso de desarrollo de software, combinado con el “Lenguaje Unificado de Modelado” (Unified Modeling Language (UML)), hacen que esta metodología sea la más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

### **5.2 Requisitos del sistema**

#### **5.2.1 Especificación de los requisitos funcionales**

A continuación, se detalla la lista con los requisitos funcionales detectados en la construcción del sistema. La base para la elaboración de estos requisitos ha sido el cliente oficial de Moodle implementado para los teléfonos iPhone. Tras un análisis de la funcionalidad implementada por

este, se concluye que aprovecha al máximo el conjunto de funciones ofrecidas por el servicio web 'Mobile Web Service' implementado a partir de la versión 2 de la plataforma.

- **Requisitos funcionales de la gestión de usuarios:**

- Login en cualquier Moodle que tenga los "Mobile Web Services" activos. El sistema deberá poder hacer login en cualquier servidor Moodle que tenga los servicios web para las plataformas móviles activos. Para ello, el sistema deberá mostrar un formulario donde el usuario tendrá que introducir la url del servidor, su nombre de usuario y su contraseña. La aplicación deberá obtener un token de acceso para los servicios web e indicar al usuario si existe algún problema de conexión o si las credenciales aportadas no son válidas en el servidor especificado.
- Posibilidad de hacer login con varias cuentas de usuario. El sistema deberá permitir que podamos logarnos con varias cuentas usuario sin necesidad de hacer login cada una de las veces. Para ello, se mostrará una lista con las cuentas de usuario almacenadas.
- Eliminar cuentas de usuario almacenadas. El sistema deberá permitir eliminar las cuentas de usuario que ya no necesitamos. Para llevar la acción a cabo, al acceder a la lista de cuentas almacenadas, si dejamos pulsado nuestro dedo sobre la cuenta que queremos eliminar, aparecerá la opción de borrado.

- **Requisitos funcionales de la subida de contenidos:**

- Subida de fotos almacenadas en el teléfono. El sistema deberá permitir que el usuario pueda explorar las fotos que tiene almacenadas en el teléfono y ofrecerle la posibilidad de subirlas al espacio personal que tiene asignado en el sistema.
- Hacer fotos con el teléfono y subirlas al sistema. El sistema deberá permitir que el usuario pueda usar la cámara de su teléfono para tomar una foto y, una vez la haya tomado, darle la posibilidad de subirla a su espacio personal asignado en el sistema.
- Grabar un fichero de audio con el teléfono y subirlo al sistema. El sistema deberá permitir que el usuario pueda grabar un fichero de audio y, una vez lo haya grabado, darle la posibilidad de subirlo a su espacio personal asignado en el sistema.

- **Requisitos funcionales de la gestión de participantes:**

- Listar los participantes de cada curso visible por el usuario. El sistema deberá permitir que el usuario pueda ver la lista de participantes de cada curso a los que tiene acceso. Desde el menú principal de la aplicación, podrá acceder a los participantes que se encontrarán agrupados por cursos.
- Visualizar la información de un usuario detalladamente. El sistema deberá permitir que el usuario pueda acceder a la información detallada (nombre, apellidos, dirección, email...) de los usuarios de los cursos a los que tiene acceso.
- Enviar un mensaje privado a un usuario. El sistema deberá permitir que el usuario pueda mandar un mensaje privado a los usuarios de los cursos a los que tiene acceso.
- Añadir un usuario a los contactos del teléfono. El sistema deberá permitir que el usuario pueda importar a la libreta de direcciones del teléfono a los usuarios de los cursos a los que tiene acceso.
- Añadir una nota para un usuario. Añadir una nota para un usuario. La nota es de texto y esta funcionalidad solo está disponible para aquellos usuarios que tengan el perfil profesor.

- **Requisitos funcionales de la gestión de cursos:**

- Listar los cursos que tiene accesibles el usuario. El sistema deberá permitir que el usuario pueda ver la lista de cursos a los que tiene acceso. Dicho acceso estará disponible desde el menú principal de la aplicación.
- Acceder al detalle de un curso. El sistema deberá permitir que el usuario pueda acceder al detalle de los cursos a los que tiene acceso, incluyendo todas sus secciones, los módulos que componen cada sección y todos los contenidos que incluye cada módulo.

- **Requisitos funcionales de la gestión de contenidos:**

- Descargar al teléfono los contenidos de tipo archivo. El sistema deberá permitir que el usuario pueda descargar al teléfono los contenidos de tipo archivo (Excel, Word, imágenes, pdf...) de los cursos a los que tiene acceso.
- Visualizar los contenidos de tipo “archivo”. El sistema deberá permitir que el usuario pueda visualizar los contenidos de tipo “archivo” siempre y cuando el teléfono disponga de un visor apropiado para el fichero que se intenta visualizar. Generalmente, se podrán abrir la mayoría de archivos ya que los teléfonos Android tienen visores de archivos para los formatos más comunes.

- Acceder a los contenidos de tipo “URL”. El sistema deberá permitir que el usuario pueda acceder a las direcciones web de los módulos tipo “Url” definidos en el detalle de un curso. Además, dicho acceso se realizará con un navegador propio de la aplicación con las opciones básicas (atrás, adelante, refrescar...).
- Abrir los contenidos de tipo “URL” desde el navegador del teléfono. El sistema deberá permitir que el usuario pueda abrir los contenidos de tipo “Url” desde el navegador del teléfono (aparte de REQ-016, que se abre desde el navegador de la aplicación).
- Listar el contenido de tipo “carpeta”. El sistema deberá permitir que el usuario pueda listar el contenido de tipo “carpeta”.
- Bajar/visualizar el contenido de tipo “carpeta”. El sistema deberá permitir que el usuario pueda bajar/visualizar el contenido de tipo “carpeta”.
- Acceder a la versión web de Moodle. El sistema deberá permitir que el usuario pueda acceder a la versión adaptada para móviles de Moodle desde un navegador propio de la aplicación.
- Acceder a la ayuda de Moodle. El sistema deberá permitir que el usuario pueda acceder a la ayuda de Moodle desde un navegador propio de la aplicación.

- **Requisitos funcionales de las gráficas estadísticas:**

- El usuario podrá obtener gráficas estadísticas que muestren su actividad para un sitio Moodle en concreto, desde el terminal por el que accede. Esa actividad incluye número de logins, ficheros subidos y accesos a cursos.

## 5.2.2 Especificación de los requisitos de almacenamiento

Al igual que en el punto anterior, en este apartado se expondrán una serie de requisitos pero esta vez relacionados con los datos que deberá almacenar el sistema para su correcto funcionamiento:

- Almacenar cuentas de acceso. El sistema deberá almacenar las diferentes cuentas de acceso con las que se ha hecho login, incluyendo url del servidor Moodle, nombre de usuario y contraseña.
- Almacenar actividad de usuario. El sistema deberá almacenar toda la actividad relativa a un usuario. Dicha actividad incluye número de logins, subida de ficheros y acceso a cursos.

### 5.2.3 Especificación de los requisitos de rendimiento

Hay que evitar hacer un número elevado a los web Services de Moodle, ya que ello puede provocar lentitud en el funcionamiento de nuestra aplicación. Si el terminal se encuentra conectado a la red wifi, la caída del rendimiento por las múltiples conexiones a los web Services no se notará demasiado. Sin embargo, cuando el terminal se encuentre conectado a una red 3G, y especialmente si no dispone de demasiada cobertura, las conexiones http penalizarán mucho la sensación del usuario, pudiendo incluso fallar aleatoriamente la carga los Activities, debido a que se produciría un Timeout en la conexión con el servidor.

Por tanto, en nuestra aplicación vamos a optar por cachear algunos de los elementos que provienen de Moodle, ya que, en general, estos no son muy cambiantes y con ello conseguiremos una mayor fluidez en la aplicación.

Los requisitos detectados y que podrían aumentar el rendimiento de la aplicación son:

- Cacheo de cursos. El sistema realizará un cacheo de los cursos obtenidos a través de los servicios web de Moodle. Los datos permanecerán en local en el teléfono durante un tiempo que se puede cambiar en las constantes de la aplicación. Actualmente, el valor de la constante es de 5 minutos, de tal forma que si se acude a los servicios web de Moodle para consultar los cursos y se vuelve a realizar la misma petición, por ejemplo, transcurridos 3 minutos desde la primera, el sistema utilizará los datos locales en vez de acudir de nuevo al servidor.
- Cacheo de participantes en un curso. El sistema realizará un cacheo de los participantes de los cursos a los que el usuario tiene acceso, obtenidos a través de los servicios web de Moodle. Los datos permanecerán en local en el teléfono durante un tiempo que se puede cambiar en las constantes de la aplicación. Actualmente, el valor de la constante es de 5 minutos, de tal forma que si se acude a los servicios web de Moodle para consultar los cursos y los participantes de los mismos, y se vuelve a realizar la misma petición, por ejemplo, transcurridos 3 minutos desde la primera, el sistema utilizará los datos locales en vez de acudir de nuevo al servidor.
- Cacheo de iconos de los módulos pertenecientes a un curso. El sistema realizará un cacheo de los iconos de los módulos pertenecientes al detalle de un curso. Nos descargaremos todos los iconos posibles y en vez de hacer llamadas a Moodle para descargar el icono (en el json de vuelta sólo viene la url del icono), utilizamos los recursos locales.



### 5.2.4 Especificación de los requisitos de idioma

Al tratarse de un cliente genérico para Moodle, la aplicación debería estar disponible en varios idiomas para que pudieran hacer uso de ella personas de distintas nacionalidades. El objetivo consiste en poder adaptarla y traducirla a otros idiomas sin tener que cambiar el código fuente de la misma. De este modo estaríamos hablando de una aplicación multiidioma.

Los idiomas disponibles serán español (por defecto), inglés y francés ya que con ellos abarcaríamos a un amplio número de usuarios. La aplicación se adaptará al idioma en el que se encuentre el sistema operativo del terminal sobre el que se encuentra instalada. En caso de no estar disponible, se mostrará en español.

## 5.3 Análisis del sistema

### 5.3.1 Modelos de casos de uso

En este punto, se realizará una descripción más detallada de los requisitos funcionales mediante la técnica de los casos de uso. Estos indican la secuencia de pasos que deben seguir los actores así como la interacción del sistema con estos.

#### 5.3.1.1 Casos de uso destinados a Login

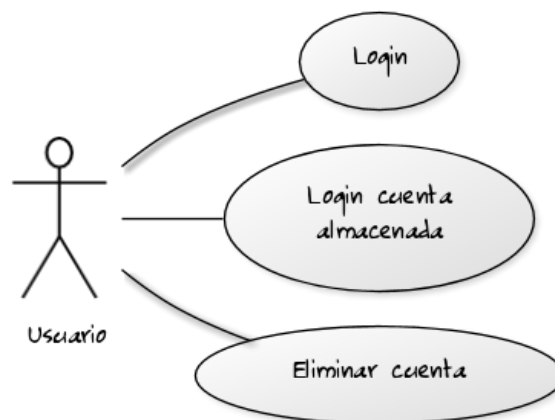


Figura 5.1: Diagrama de casos de uso - Login

- **Login**

- **Descripción:** login de un usuario.
- **Precondición:** el usuario está dado de alta en el sitio Moodle.
- **Postcondición:** el usuario accede al sitio Moodle.
- **Actores:** usuario.
- **Resumen:** cuando el usuario arranca la aplicación, debe indicar al sitio Moodle que se quiere conectar, así como las credenciales de acceso.
- **Escenario principal:**
  1. El caso de uso comienza cuando el usuario ejecuta la aplicación o cuando quiere iniciar una nueva sesión.
  2. El usuario introduce la url del sitio al que quiere acceder, su nombre y contraseña.
  3. El sistema hace login con los parámetros introducidos.
- **Escenario alternativo:**
  0. En cualquier momento, el usuario puede salir del proceso de login.
  3. Si se deja alguno de los campos sin rellenar, si no se puede conectar con el servidor de Moodle o si las credenciales de acceso son incorrectas, el sistema muestra un mensaje de error.

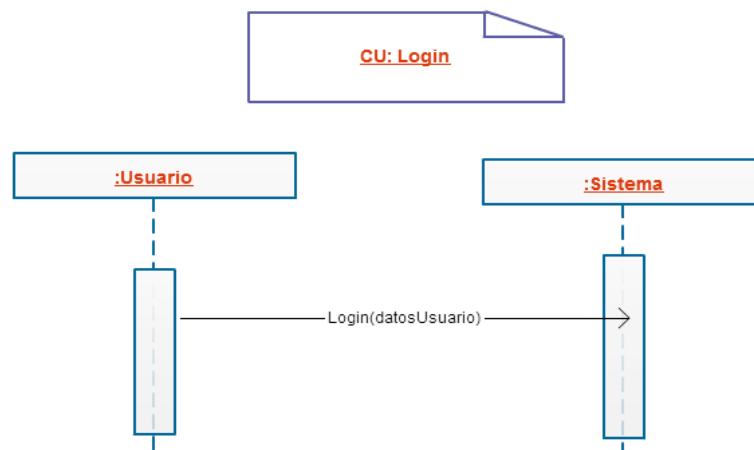


Figura 5.2: Diagrama de secuencia – Login

**Contrato de la operación:** Login(datosUsuario)

- **Responsabilidades:** acceso al sitio Moodle especificado.
- **Referencias cruzadas:** Caso de uso “Login”.
- **Precondición:** datosUsuario es válido
- **Postcondición:** acceso al sistema. En caso de no existir, se guarda al usuario en el almacén de datos.

- **Acceso desde cuenta almacenada**

- **Descripción:** login de un usuario desde una cuenta almacenada.
- **Precondición:** ya se ha accedido previamente con dichas credenciales.
- **Postcondición:** el usuario accede al sitio Moodle.
- **Actores:** usuario.
- **Resumen:** cuando el usuario arranca la aplicación, puede acceder a un listado de cuentas almacenadas y seleccionar una de ellas.
- **Escenario principal:**
  1. El caso de uso comienza cuando el usuario ejecuta la aplicación o cuando quiere iniciar una nueva sesión.
  2. El usuario accede al listado de cuentas almacenadas.
  3. Tras la selección de una de las cuentas, el sistema hace login.
- **Escenario alternativo:**
  0. En cualquier momento, el usuario puede salir del proceso de login.
  2. Si no existen cuentas almacenadas, se mostrará una lista vacía.
  3. Si las credenciales de acceso son incorrectas, el sistema muestra un mensaje de error.

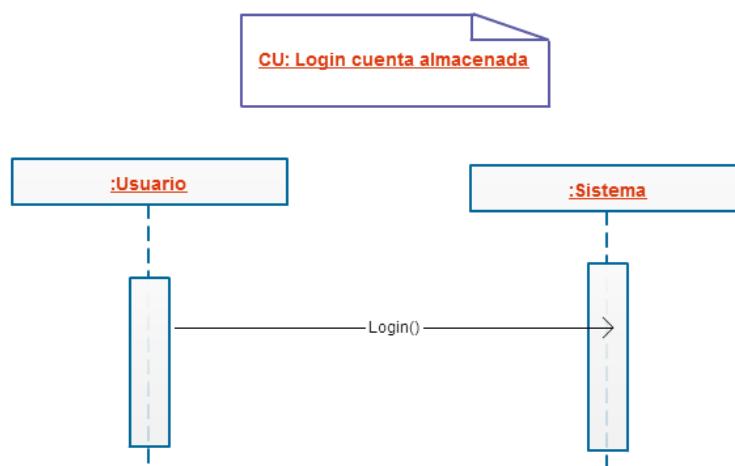


Figura 5.3: Diagrama de secuencia – Login cuenta almacenada

**Contrato de la operación: Login()**

- **Responsabilidades:** acceso al sitio Moodle.
- **Referencias cruzadas:** Caso de uso “Login cuenta almacenada”.
- **Precondición:** existe al menos una cuenta almacenada.
- **Postcondición:** acceso al sistema.

- **Eliminar una cuenta almacenada**

- **Descripción:** borrado de una cuenta almacenada.
- **Precondición:** ya se ha accedido previamente con dichas credenciales.
- **Postcondición:** se elimina la cuenta seleccionada.
- **Actores:** usuario
- **Resumen:** cuando el usuario arranca la aplicación, puede acceder a un listado de cuentas almacenadas y seleccionar una de ellas para eliminarla.
- **Escenario principal:**
  1. El caso de uso comienza cuando el usuario ejecuta la aplicación o cuando quiere iniciar una nueva sesión.
  2. El usuario accede al listado de cuentas almacenadas.
  3. Tras la selección de una de las cuentas para ser eliminada, el sistema la borra (se requerirá confirmación).
- **Escenario alternativo:**
  0. En cualquier momento, el usuario puede salir del proceso de login.
  3. Si se produce un error a la hora de borrar la cuenta, se mostrará un mensaje de error.

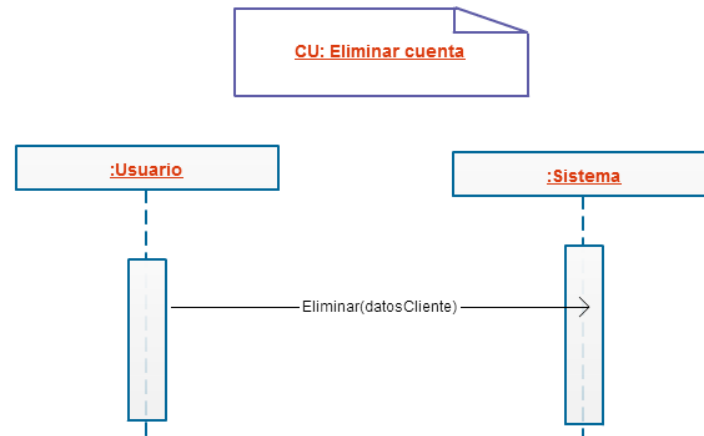


Figura 5.4: Diagrama de secuencia – Eliminar cuenta

**Contrato de la operación:** Eliminar(datosCliente)

- **Responsabilidades:** elimina una cuenta de acceso.
- **Referencias cruzadas:** Caso de uso “Eliminar cuenta almacenada”.
- **Precondición:** existe al menos una cuenta almacenada.
- **Postcondición:** se elimina la cuenta del almacén de datos.

### 5.3.1.2 Casos de uso destinados a la subida de contenidos

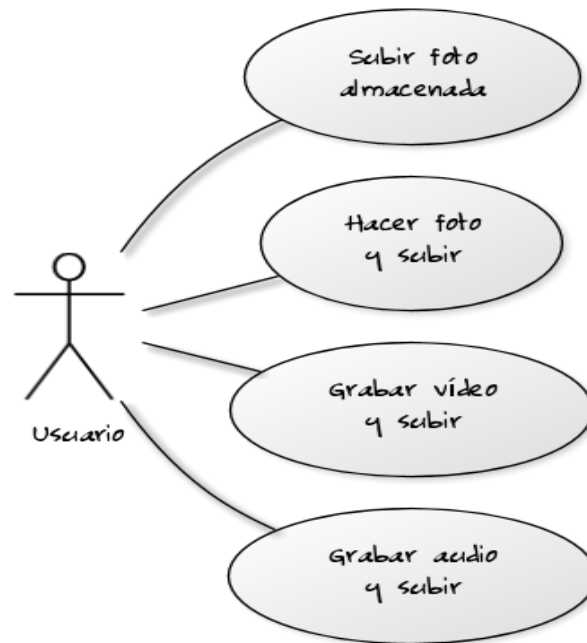


Figura 5.5: Diagrama de casos de uso – Contenidos

- **Subir foto almacenada en el terminal**
  - **Descripción:** subida de una foto almacenada.
  - **Precondición:** el usuario se encuentra logado.
  - **Postcondición:** se subirá la foto al espacio privado del usuario.
  - **Actores:** usuario.
  - **Resumen:** desde el menú de Subidas, el usuario elegirá la opción de subir una foto y tras seleccionar la imagen deseada, la subirá a su espacio privado.
  - **Escenario principal:**
    1. El caso de uso comienza cuando se accede al menú de Subidas.
    2. Dentro del menú, el usuario elige Buscar una foto en álbum.
    3. Se mostrará la galería de imágenes del teléfono y el usuario deberá seleccionar la imagen que desea subir.
    4. Una vez seleccionada la imagen, pulsar sobre el botón de Subir.
  - **Escenario alternativo:**
    2. El usuario puede volver al menú Principal.
    3. El usuario puede volver sin seleccionar ninguna imagen.

4. Una vez la imagen se encuentra seleccionada, el usuario puede volver sin subirla.

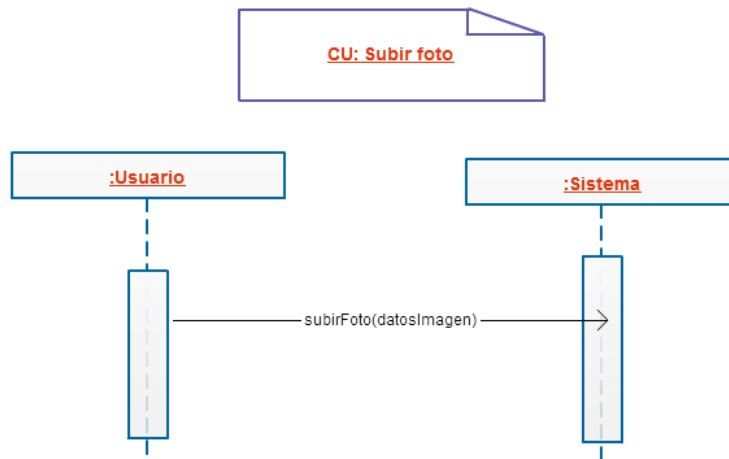


Figura 5.6: Diagrama de secuencia – Subir foto

**Contrato de la operación:** subirFoto(datosImagen)

- **Responsabilidades:** sube al espacio privado la imagen seleccionada.
- **Referencias cruzadas:** Caso de uso “Subir foto”.
- **Precondición:** existe datosImagen.
- **Postcondición:** se sube la imagen al espacio privado del usuario.

- **Hacer una foto con el terminal y subirla al espacio privado**

- **Descripción:** hacer una foto y subirla.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se subirá la foto que se acaba de tomar al espacio privado del usuario.
- **Actores:** usuario.
- **Resumen:** desde el menú de Subidas, el usuario elegirá la opción de Realizar una foto o vídeo, que posteriormente podrá subir a su espacio privado.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Subidas.
  2. Dentro del menú, el usuario elige Realizar una foto o vídeo.
  3. Se mostrará la cámara del terminal para que el usuario realice la foto.
  4. Una vez realizada la foto, pulsar sobre el botón de Subir.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal.
  3. El usuario puede volver sin realizar ninguna foto.

4. Con la foto ya tomada, el usuario puede volver sin subirla.

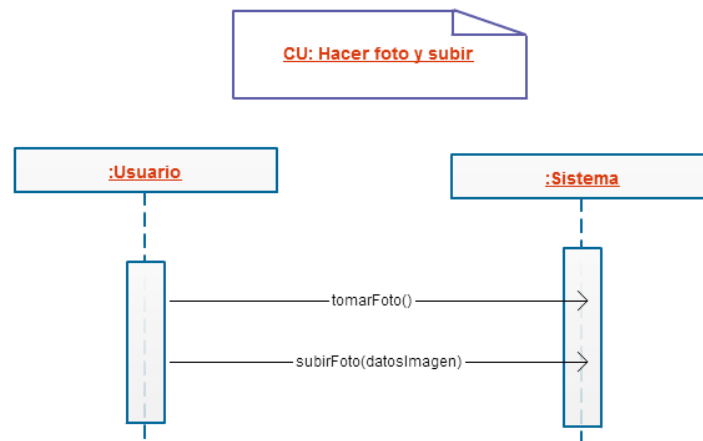


Figura 5.7: Diagrama de secuencia – Hacer foto y subirla

**Contrato de la operación: tomarFoto()**

- **Responsabilidades:** realiza una foto con la cámara del terminal.
- **Referencias cruzadas:** Caso de uso “Hacer foto y subir”.
- **Precondición:** el terminal posee cámara.
- **Postcondición:** se obtiene la foto.

- **Grabar un vídeo con el terminal y subirlo al espacio privado**

- **Descripción:** grabar un vídeo y subirlo.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se subirá el vídeo que se acaba de grabar al espacio privado del usuario.
- **Actores:** usuario.
- **Resumen:** desde el menú de Subidas, el usuario elegirá la opción de Realizar una foto o vídeo, que posteriormente podrá subir a su espacio privado.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Subidas.
  2. Dentro del menú, el usuario elige Realizar una foto o vídeo.
  3. Se mostrará la cámara del terminal para que el usuario grabe el vídeo.
  4. Una vez grabado el vídeo, pulsar sobre el botón de Subir.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal.
  3. El usuario puede volver sin grabar ningún vídeo.

4. Con el vídeo ya grabado, el usuario puede volver sin subirlo.

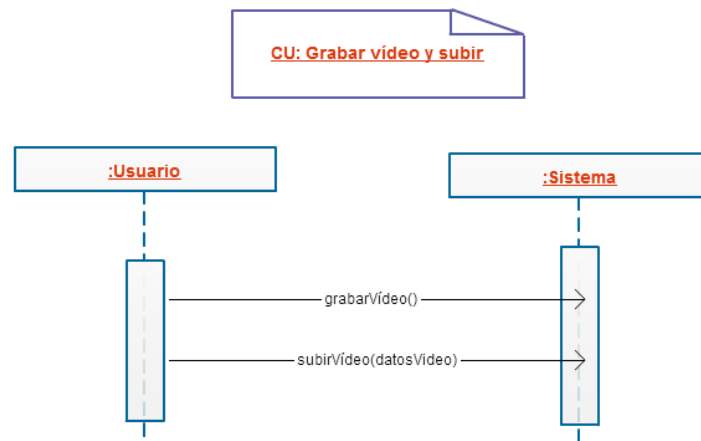


Figura 5.8: Diagrama de secuencia – Grabar vídeo y subirlo

**Contrato de la operación: grabarVideo()**

- **Responsabilidades:** grabar un vídeo con la cámara del terminal.
- **Referencias cruzadas:** Caso de uso “Grabar vídeo y subir”.
- **Precondición:** el terminal posee cámara.
- **Postcondición:** se graba el vídeo.

**Contrato de la operación: subirVideo(datosVideo)**

- **Responsabilidades:** sube al espacio privado el vídeo seleccionada.
- **Referencias cruzadas:** Caso de uso “Grabar vídeo y subir”.
- **Precondición:** existe datosVideo.
- **Postcondición:** se sube el vídeo al espacio privado del usuario.

• **Grabar un audio con el terminal y subirlo al espacio privado**

- **Descripción:** grabar un audio y subirlo.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se subirá el audio que se acaba de grabar al espacio privado del usuario.
- **Actores:** usuario.
- **Resumen:** desde el menú de Subidas, el usuario elegirá la opción de Grabar audio, que posteriormente podrá subir a su espacio privado.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Subidas.



2. Dentro del menú, el usuario elige Grabar audio.
  3. Se mostrará una pantalla para que el usuario grabe el audio.
  4. Una vez grabado el audio, pulsar sobre el botón de Subir.
- **Escenario alternativo:**
    2. El usuario puede volver al menú Principal.
    3. El usuario puede volver sin grabar ningún audio.
    4. Con el audio ya grabado, el usuario puede volver sin subirlo.

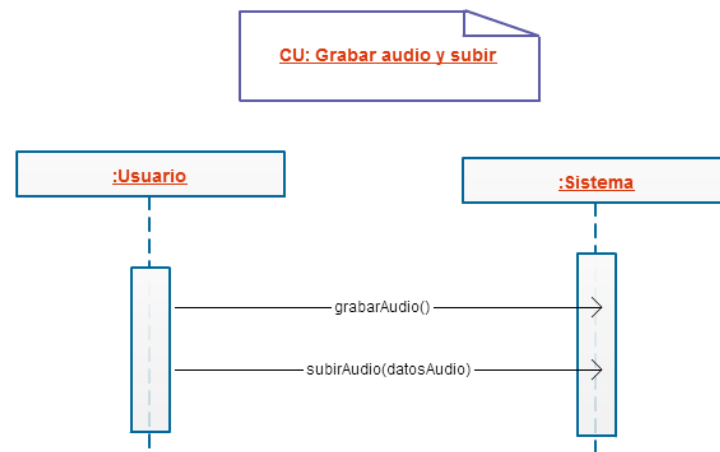


Figura 5.9: Diagrama de secuencia – Grabar audio y subirlo

**Contrato de la operación:** grabarAudio()

- **Responsabilidades:** grabar un fichero de audio con el terminal.
- **Referencias cruzadas:** Caso de uso “Grabar audio y subir”.
- **Precondición:** el terminal posee grabadora.
- **Postcondición:** se graba el audio.

**Contrato de la operación:** subirAudio(datosAudio)

- **Responsabilidades:** sube al espacio privado el audio seleccionado.
- **Referencias cruzadas:** Caso de uso “Grabar audio y subir”.
- **Precondición:** existe datosAudio.
- **Postcondición:** se sube el audio al espacio privado del usuario.

### 5.3.1.3 Casos de uso destinados a la gestión de participantes

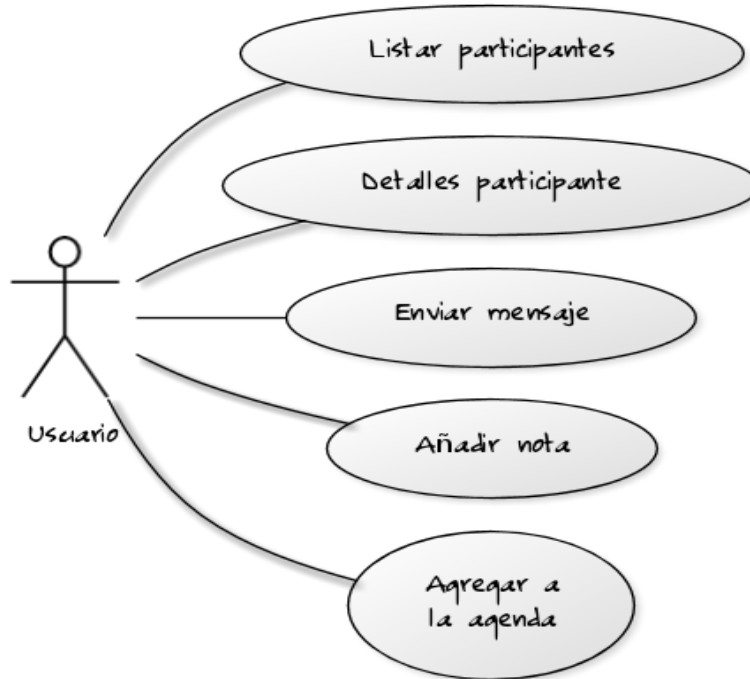


Figura 5.10: Diagrama de casos de uso – Participantes

- **Listar participantes**
  - **Descripción:** listar participantes de un curso.
  - **Precondición:** el usuario se encuentra logado.
  - **Postcondición:** se listarán todos los participantes de un curso.
  - **Actores:** usuario.
  - **Resumen:** desde el menú de Participantes, el usuario seleccionará el curso del que quiere listar los participantes.
  - **Escenario principal:**
    1. El caso de uso comienza cuando se accede al menú de Participantes.
    2. Se mostrará una lista con los cursos disponibles para el usuario.
    3. Cuando se selecciona el curso, aparecen todos los participantes que lo componen.
  - **Escenario alternativo:**
    2. El usuario puede volver al menú Principal.

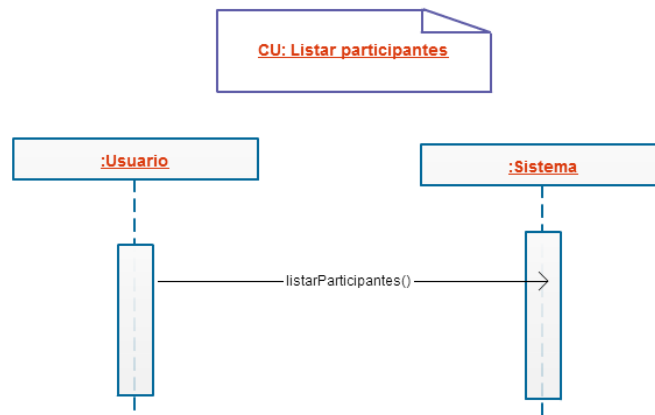


Figura 5.11: Diagrama de secuencia – Listar participantes

**Contrato de la operación:** listarParticipantes()

- **Responsabilidades:** listar los participantes de un curso.
- **Referencias cruzadas:** Caso de uso “Listar participantes”.
- **Precondición:** el usuario tiene asignado algún curso.
- **Postcondición:** se muestra el resto de participantes del curso.

• **Mostrar información detallada de un participante**

- **Descripción:** datos de un participante.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se mostrará la información relativa a un participante.
- **Actores:** usuario.
- **Resumen:** desde el menú de Participantes, el usuario seleccionará el curso del que quiere listar los participantes y elegirá a uno de ellos.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Participantes.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. Cuando se selecciona el curso, aparecen todos los participantes que lo componen.
  4. El usuario pulsa sobre el participante y se muestra su información detallada.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal.
  3. El usuario puede volver a la lista de cursos sin seleccionar un participante.

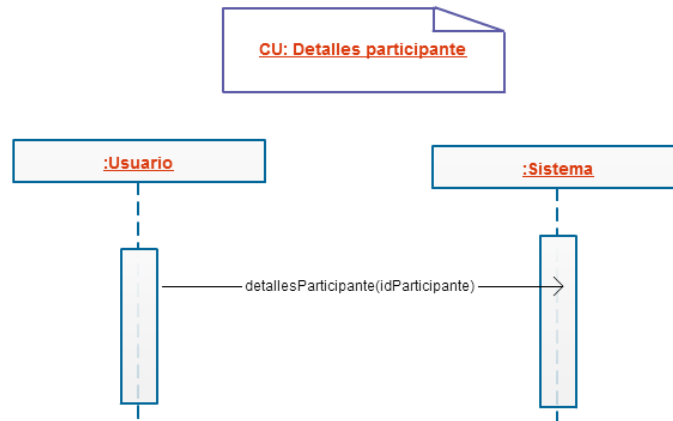


Figura 5.12: Diagrama de secuencia – Detalles participante

**Contrato de la operación:** detallesParticipante(idParticipante)

- **Responsabilidades:** mostrar información detallada del participante.
- **Referencias cruzadas:** Caso de uso “Detalles participante”.
- **Precondición:** el usuario tiene asignado algún curso y en el curso hay más participantes
- **Postcondición:** se muestra información del participante seleccionado.

• **Enviar mensaje a un participante**

- **Descripción:** enviar un mensaje a uno de los participantes.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se enviará el mensaje al participante.
- **Actores:** usuario.
- **Resumen:** desde el menú de Participantes, el usuario seleccionará el curso del que quiere listar los participantes y elegirá a uno de ellos para enviarle un mensaje.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Participantes.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. Cuando se selecciona el curso, aparecen todos los participantes que lo componen.
  4. El usuario pulsa sobre el participante y se muestra su información detallada.
  5. Si pulsa sobre el botón de enviar mensaje, se mostrará el cuadro para escribir el texto y la opción de efectuar el envío.

- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal.
  3. El usuario puede volver a la lista de cursos sin seleccionar un participante.
  4. El usuario puede volver a la lista de participantes.
  5. El usuario puede volver a la información detallada del participante sin enviar el mensaje.

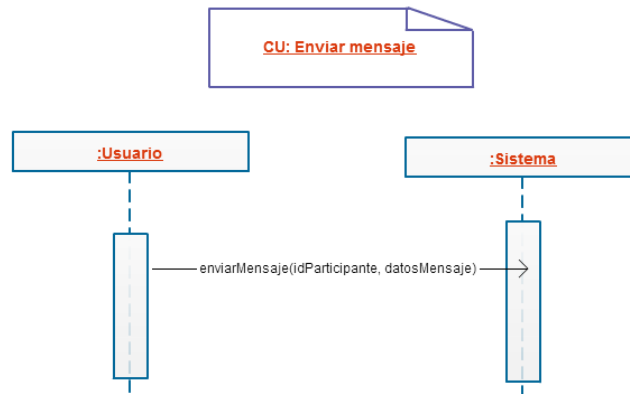


Figura 5.13: Diagrama de secuencia – Enviar mensaje

**Contrato de la operación:** enviarMensaje(idParticipante, datosMensaje)

- **Responsabilidades:** enviar un mensaje a un participante.
  - **Referencias cruzadas:** Caso de uso “Enviar mensaje”.
  - **Precondición:** el usuario tiene asignado algún curso y en el curso hay más participantes.
  - **Postcondición:** se envía el mensaje al participante.
- **Añadir nota a un participante**
    - **Descripción:** añadir una nota a uno de los participantes.
    - **Precondición:** el usuario se encuentra logado.
    - **Postcondición:** se añadirá la nota al participante.
    - **Actores:** usuario.
    - **Resumen:** desde el menú de Participantes, el usuario seleccionará el curso del que quiere listar los participantes y elegirá a uno de ellos para añadirle una nota.
    - **Escenario principal:**
      1. El caso de uso comienza cuando se accede al menú de Participantes.
      2. Se mostrará una lista con los cursos disponibles para el usuario.

3. Cuando se selecciona el curso, aparecen todos los participantes que lo componen.
  4. El usuario pulsa sobre el participante y se muestra su información detallada.
  5. Si pulsa sobre el botón de añadir nota, se mostrará el cuadro para escribir el texto y la opción de agregarla.
- **Escenario alternativo:**
    2. El usuario puede volver al menú Principal.
    3. El usuario puede volver a la lista de cursos sin seleccionar un participante.
    4. El usuario puede volver a la lista de participantes.
    5. El usuario puede volver a la información detallada del participante sin añadir la nota.

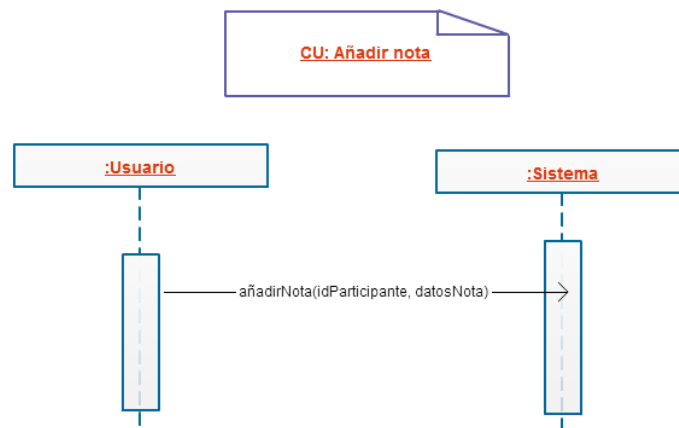


Figura 5.14: Diagrama de secuencia – Añadir nota

**Contrato de la operación:** añadirNota(idParticipante, datosNota)

- **Responsabilidades:** añadir una nota a un participante.
  - **Referencias cruzadas:** Caso de uso “Añadir nota”.
  - **Precondición:** el usuario tiene asignado algún curso y en el curso hay más participantes.
  - **Postcondición:** se añade la nota al participante.
- **Agregar un participante a la agenda del terminal**
    - **Descripción:** agregar participante como contacto del terminal.
    - **Precondición:** el usuario se encuentra logado.
    - **Postcondición:** se añadirá al participante a la lista de contactos.
    - **Actores:** usuario.

- **Resumen:** desde el menú de Participantes, el usuario seleccionará el curso del que quiere listar los participantes y elegirá a uno de ellos para agregarlo a la lista de contactos.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Participantes.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. Cuando se selecciona el curso, aparecen todos los participantes que lo componen.
  4. El usuario pulsa sobre el participante y se muestra su información detallada.
  5. Si pulsa sobre el botón de añadir a contactos, se agregará a los contactos del terminal.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal.
  3. El usuario puede volver a la lista de cursos sin seleccionar un participante.
  4. El usuario puede volver a la lista de participantes.

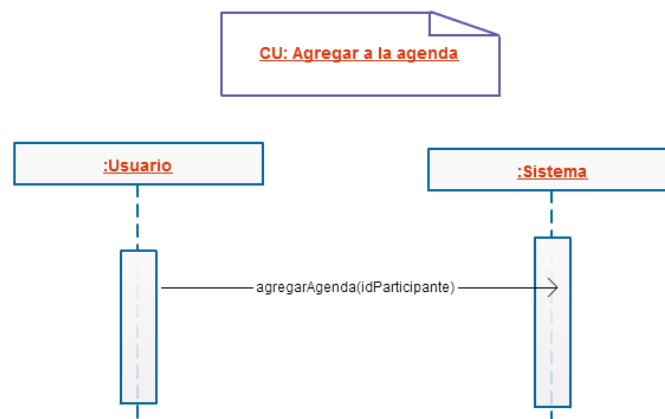


Figura 5.15: Diagrama de secuencia – Agregar a la agenda

**Contrato de la operación:** `agregarAgenda(idParticipante)`

- **Responsabilidades:** agregar a un participante a la agenda del terminal.
- **Referencias cruzadas:** Caso de uso “Agregar a la agenda”.
- **Precondición:** el usuario tiene asignado algún curso y en el curso hay más participantes.
- **Postcondición:** se agrega al participante en la lista de contactos del terminal.

### 5.3.1.4 Casos de uso destinados a la gestión de contenidos

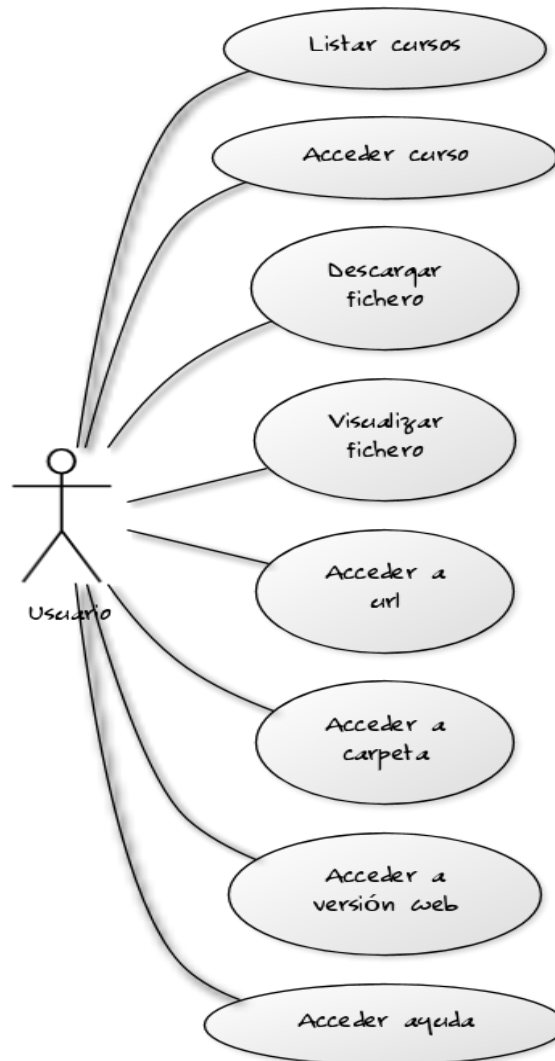


Figura 5.16: Diagrama de casos de uso – Cursos

- **Listar cursos**
  - **Descripción:** listar cursos accesibles para el usuario.
  - **Precondición:** el usuario se encuentra logado.
  - **Postcondición:** se listarán todos los cursos disponibles.
  - **Actores:** usuario.
  - **Resumen:** desde el menú Principal, el usuario accederá a la opción de Cursos.
  - **Escenario principal:**
    1. El caso de uso comienza cuando se accede al menú de Cursos.
    2. Se mostrará una lista con los cursos disponibles para el usuario.
  - **Escenario alternativo:**



2. El usuario puede volver al menú Principal.

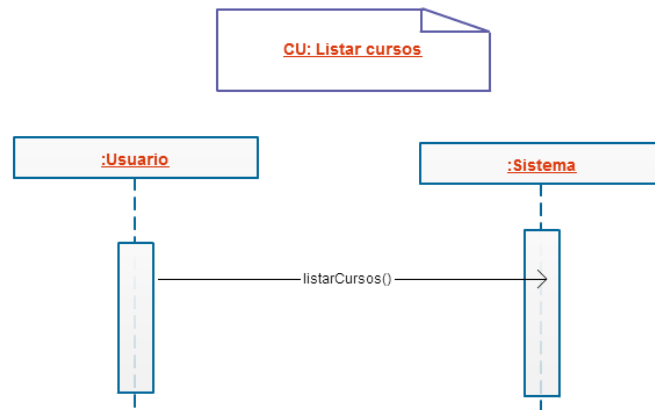


Figura 5.17: Diagrama de secuencia – Listar cursos

**Contrato de la operación:** listarCursos()

- **Responsabilidades:** listar cursos a los que pertenece el usuario.
- **Referencias cruzadas:** Caso de uso “Listar cursos”.
- **Precondición:** el usuario tiene asignado algún curso.
- **Postcondición:** se muestran todos los cursos asignados al usuario.

- **Acceder a un curso**

- **Descripción:** acceder al detalle de un curso.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se accederá al detalle de un curso.
- **Actores:** usuario.
- **Resumen:** desde el menú Principal, el usuario accederá a la opción de Cursos y seleccionará uno de los que tiene disponibles, mostrándose todas sus secciones, módulos y contenidos.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Cursos.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. El usuario seleccionará uno de los cursos.
  4. Se mostrará el detalle del curso seleccionado.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal.

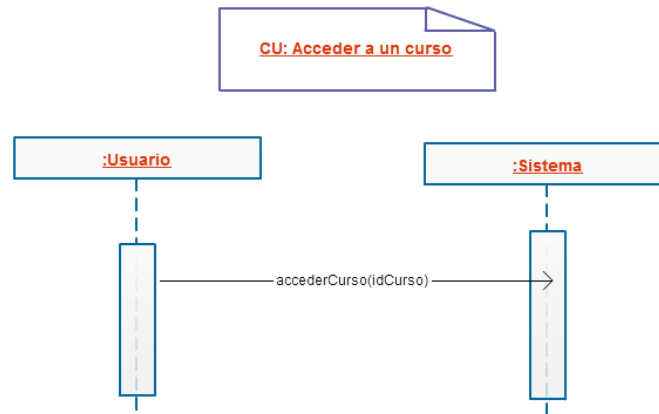


Figura 5.18: Diagrama de secuencia – Acceder curso

**Contrato de la operación:** accederCurso(idCurso)

- **Responsabilidades:** acceder al curso seleccionado.
- **Referencias cruzadas:** Caso de uso “Acceder a un curso”.
- **Precondición:** el usuario tiene asignado algún curso.
- **Postcondición:** se muestra el contenido del curso.

• **Descargar fichero**

- **Descripción:** descargar fichero perteneciente al detalle de un curso.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se descargará al terminal el fichero.
- **Actores:** usuario.
- **Resumen:** desde el menú Principal, el usuario accederá a la opción de Cursos y seleccionará uno de los que tiene disponibles, mostrándose todas sus secciones, módulos y contenidos. Elegirá uno de los ficheros para descargarlo al terminal.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Cursos.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. El usuario seleccionará uno de los cursos.
  4. Se mostrará el detalle del curso seleccionado.
  5. El usuario descargará el fichero deseado.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal
  4. El usuario puede volver a la lista de Cursos.

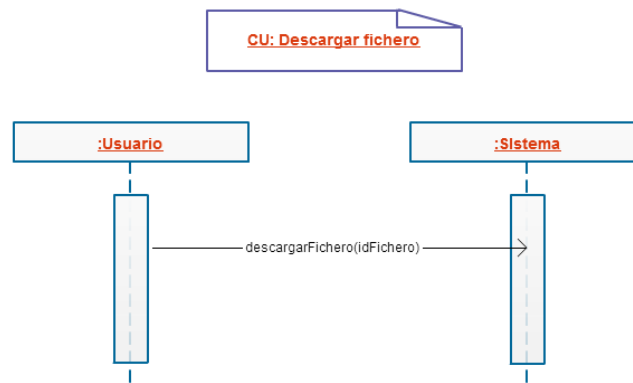


Figura 5.19: Diagrama de secuencia – Descargar fichero

**Contrato de la operación:** descargarFichero(idFichero)

- **Responsabilidades:** descargar al terminal el fichero seleccionado.
- **Referencias cruzadas:** Caso de uso “Descargar fichero”.
- **Precondición:** el usuario tiene asignado algún curso y dentro del curso existe al menos un fichero para descargar.
- **Postcondición:** se descarga el fichero al terminal.

- **Visualizar un fichero descargado**

- **Descripción:** visualizar el contenido de un fichero previamente descargado.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se visualizará el contenido del fichero.
- **Actores:** usuario.
- **Resumen:** desde el menú Principal, el usuario accederá a la opción de Cursos y seleccionará uno de los que tiene disponibles, mostrándose todas sus secciones, módulos y contenidos. Elegirá uno de los ficheros para descargarlo al terminal. Una vez descargado, lo seleccionará para mostrar su contenido.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Cursos.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. El usuario seleccionará uno de los cursos.
  4. Se mostrará el detalle del curso seleccionado.
  5. El usuario descargará el fichero deseado.
  6. El usuario visualizará el contenido.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal
  4. El usuario puede volver a la lista de Cursos.

6. No se puede abrir el fichero porque el terminal no dispone del visor adecuado para su exploración.

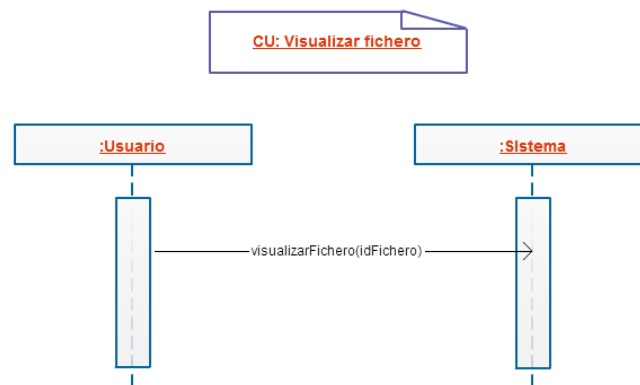


Figura 5.20: Diagrama de secuencia – Visualizar fichero

**Contrato de la operación:** visualizarFichero(idFichero)

- **Responsabilidades:** visualizar el contenido del fichero descargado.
- **Referencias cruzadas:** Caso de uso “Visualizar fichero”.
- **Precondición:** el usuario ya se ha descargado el fichero y posee en el terminal un visor para poder visualizar su contenido.
- **Postcondición:** se visualiza el contenido del fichero.

- **Acceder a una url**

- **Descripción:** acceder a una url perteneciente al detalle de un curso.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se accederá a la url.
- **Actores:** usuario.
- **Resumen:** desde el menú Principal, el usuario accederá a la opción de Cursos y seleccionará uno de los que tiene disponibles, mostrándose todas sus secciones, módulos y contenidos. Tras seleccionar un contenido de tipo url, se visualizará.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede al menú de Cursos.
  2. Se mostrará una lista con los cursos disponibles para el usuario.
  3. El usuario seleccionará uno de los cursos.
  4. Se mostrará el detalle del curso seleccionado.

5. El usuario visualizará el contenido de una url.
- **Escenario alternativo:**
  3. El usuario puede volver al menú Principal
  4. El usuario puede volver a la lista de Cursos.

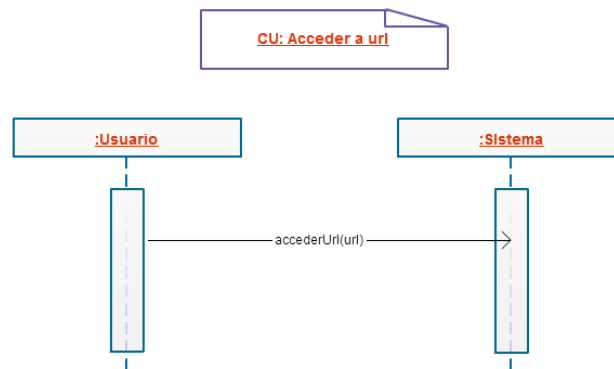


Figura 5.21: Diagrama de secuencia – Acceder Url

**Contrato de la operación: accederUrl(url)**

- **Responsabilidades:** acceder a la url seleccionada.
  - **Referencias cruzadas:** Caso de uso “Acceder a url”.
  - **Precondición:** el usuario tiene asignado algún curso y dentro del curso existe al menos un contenido de tipo url.
  - **Postcondición:** se accede al contenido del tipo url.
- **Acceder al contenido de una carpeta**
    - **Descripción:** acceder al contenido de una carpeta en el detalle de un curso.
    - **Precondición:** el usuario se encuentra logado.
    - **Postcondición:** se accederá al contenido de una carpeta.
    - **Actores:** usuario.
    - **Resumen:** desde el menú Principal, el usuario accederá a la opción de Cursos y seleccionará uno de los que tiene disponibles, mostrándose todas sus secciones, módulos y contenidos. Tras seleccionar un contenido de tipo carpeta, se mostrará.
    - **Escenario principal:**
      1. El caso de uso comienza cuando se accede al menú de Cursos.
      2. Se mostrará una lista con los cursos disponibles para el usuario.
      3. El usuario seleccionará uno de los cursos.
      4. Se mostrará el detalle del curso seleccionado.

5. El usuario explorará el contenido de una carpeta.
- **Escenario alternativo:**
  2. El usuario puede volver al menú Principal
  4. El usuario puede volver a la lista de Cursos.

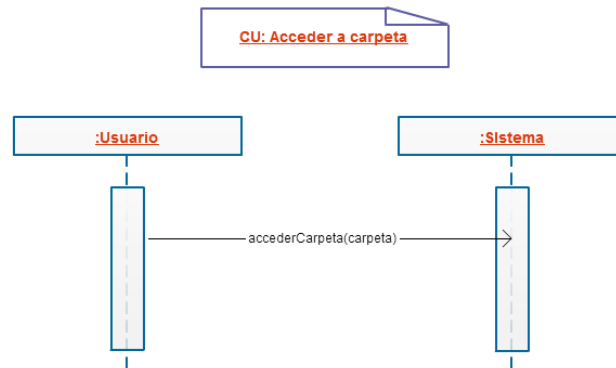


Figura 5.22: Diagrama de secuencia – Acceder carpeta

**Contrato de la operación:** accederCarpeta(carpeta)

- **Responsabilidades:** acceder a la carpeta seleccionada.
  - **Referencias cruzadas:** Caso de uso “Acceder a carpeta”.
  - **Precondición:** el usuario tiene asignado algún curso y dentro del curso existe al menos un contenido de tipo carpeta.
  - **Postcondición:** se accede al contenido del tipo carpeta.
- **Acceder a la versión web**
    - **Descripción:** acceder a la versión web del sitio Moodle.
    - **Precondición:** el usuario se encuentra logado.
    - **Postcondición:** se accederá a la versión web del sitio.
    - **Actores:** usuario.
    - **Resumen:** desde el menú Principal, el usuario accederá a la opción Web y se mostrará la versión web del sitio.
    - **Escenario principal:**
      1. El caso de uso comienza cuando se accede a la opción Web.

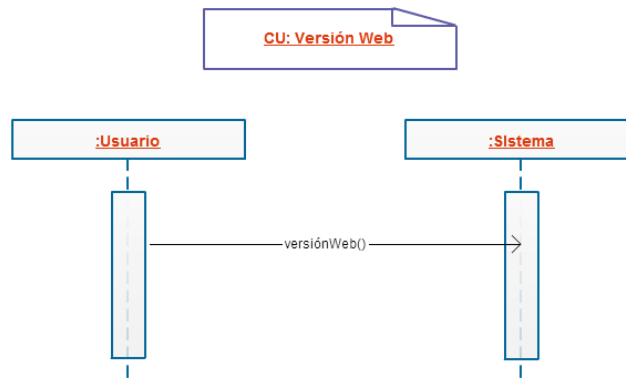


Figura 5.23: Diagrama de secuencia – Versión web

**Contrato de la operación:** versiónWeb()

- **Responsabilidades:** acceder a la versión web del sitio Moodle.
- **Referencias cruzadas:** Caso de uso “Versión Web”.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se accede a la versión web del sitio Moodle.

• **Acceder a la ayuda**

- **Descripción:** acceder a la ayuda de Moodle.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se accederá a la ayuda.
- **Actores:** usuario.
- **Resumen:** desde el menú Principal, el usuario accederá a la opción de Ayuda y se mostrará la ayuda de Moodle.
- **Escenario principal:**
  1. El caso de uso comienza cuando se accede a la opción de Ayuda.

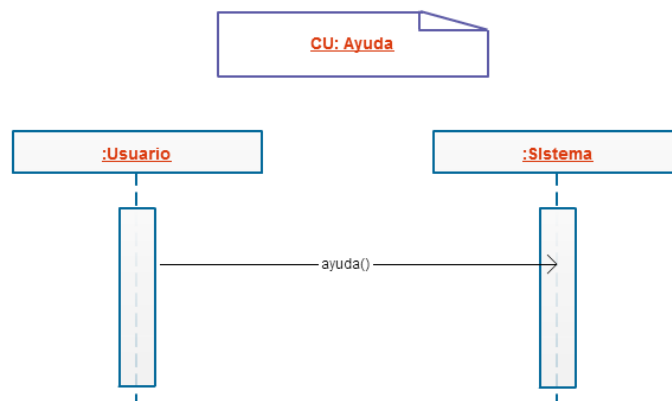


Figura 5.24: Diagrama de secuencia – Ayuda

**Contrato de la operación:** ayuda()

- **Responsabilidades:** acceder a la ayuda de Moodle.
- **Referencias cruzadas:** Caso de uso “Ayuda”.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se accede a la ayuda de Moodle.

**5.3.1.5 Casos de uso destinados a los datos estadísticos**



Figura 5.25: Diagrama de casos de uso – Estadísticas

- **Dibujar estadísticas**
  - **Descripción:** acceder a la gráfica con los datos estadísticos.
  - **Precondición:** el usuario se encuentra logado.
  - **Postcondición:** se dibuja la gráfica con los datos.
  - **Actores:** usuario.
  - **Resumen:** desde el menú Principal, el usuario accederá a la opción Estadísticas para ver sus datos.
  - **Escenario principal:**
    1. El caso de uso comienza cuando se accede al menú de Estadísticas.
    2. Se dibujará una gráfica indicando el número de logins, ficheros subidos y acceso a cursos realizados por el usuario a ese sitio desde el terminal.



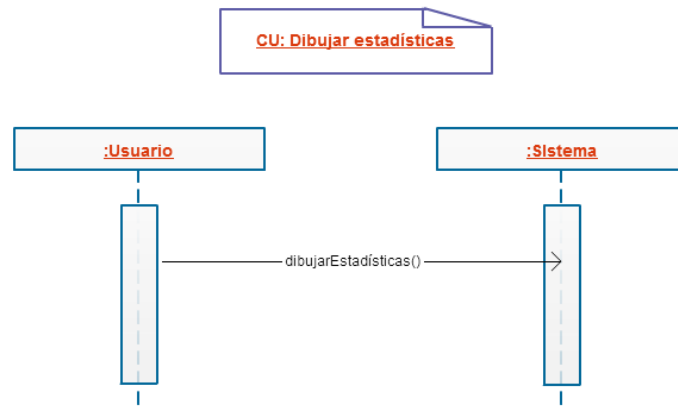


Figura 5.26: Diagrama de secuencia – Dibujar estadísticas

**Contrato de la operación:** dibujarEstadísticas()

- **Responsabilidades:** dibujar estadísticas referentes al usuario.
- **Referencias cruzadas:** Caso de uso “Dibujar estadísticas”.
- **Precondición:** el usuario se encuentra logado.
- **Postcondición:** se dibujan las estadísticas de uso del usuario logado.

## 5.3.2 Modelo conceptual de datos

### 5.3.2.1 Diagrama de clases conceptuales

En la figura 5.26 podemos ver una simplificación del diagrama conceptual de clases. Se ha optado por incluir esta versión reducida en la documentación para que su comprensión sea mucho más sencilla.

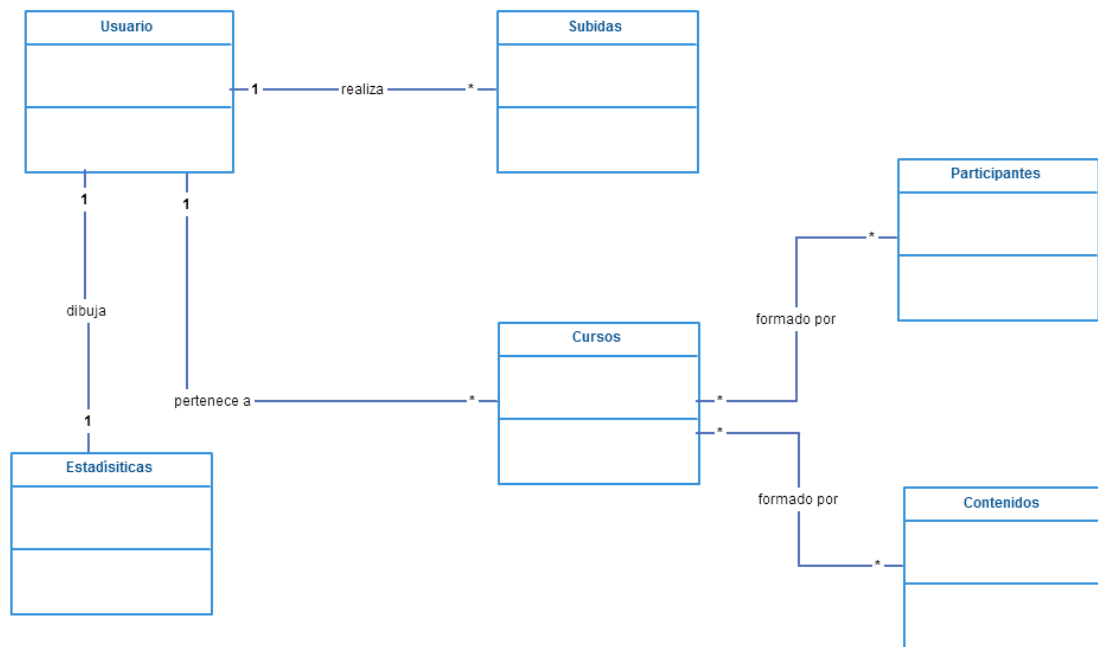


Figura 5.27: Diagrama conceptual de clases

---

### ***Diseño***

---

Para el diseño del sistema se ha seguido un patrón arquitectónico de tres capas. De este modo, podemos dividir la carga de trabajo en 3 niveles, con un reparto claro de funciones. A continuación se presentan las capas ordenadas de mayor a menor nivel de abstracción:

- Capa de presentación: es la encargada de interactuar con el usuario.
- Capa de dominio: es la encargada de implementar las funcionalidades del sistema.
- Capa de gestión de datos: es la encargada de interactuar con el “SGBD”.

Cada capa de las indicadas anteriormente sólo se comunica con su capa contigua.

A continuación se muestra un diagrama con la arquitectura propuesta para la aplicación, incluyendo las capas en las que está subdividido el software, la interacción con elementos externos y con el terminal del usuario.

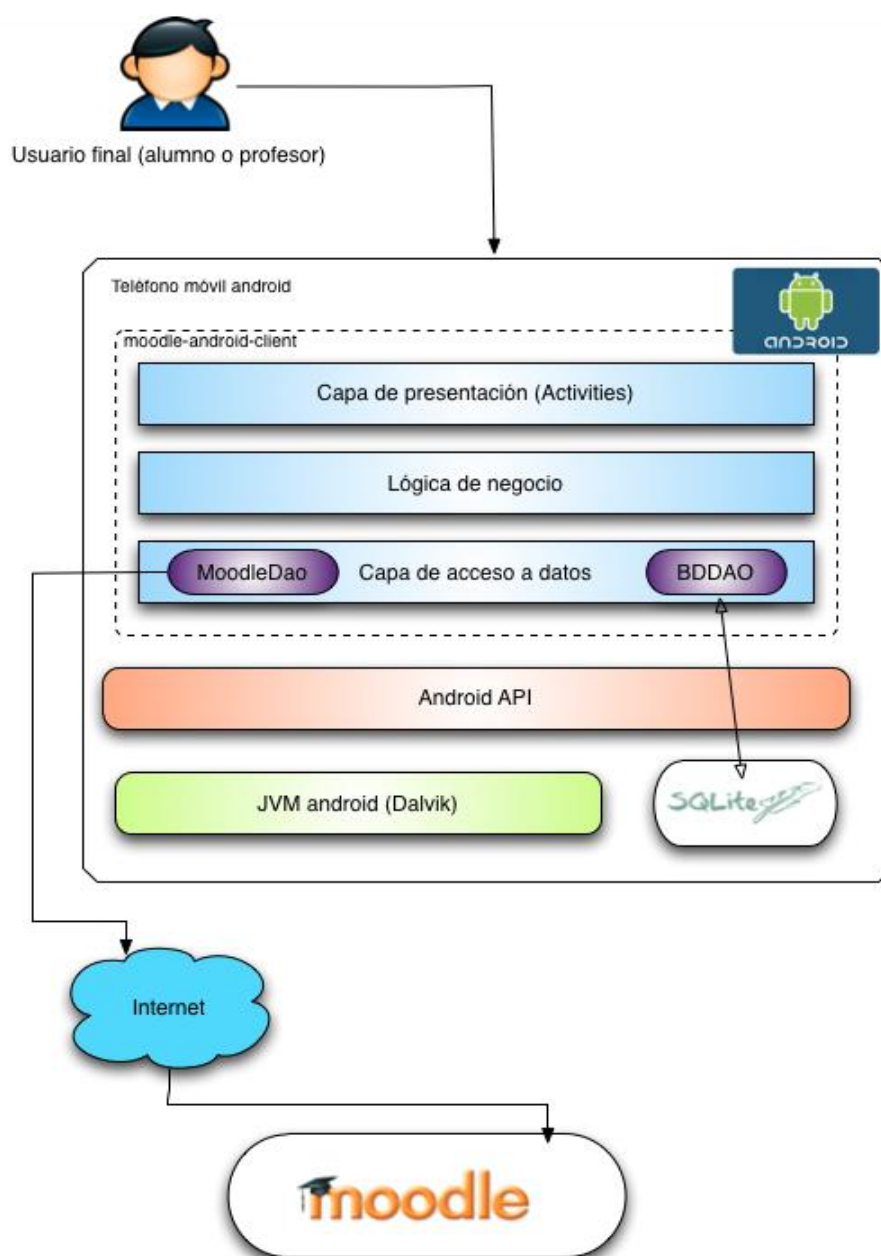


Figura 6.1: Arquitectura de nuestra aplicación

## 6.1 Estructura de un proyecto Android

Todos los proyectos Android tienen la siguiente estructura:

- **Libs:** en esta carpeta deberán ir todos los ficheros de librerías externas que utilice la aplicación. En nuestro caso, la carpeta tiene un jar para el parseo de objetos json (gson.jar) y otro para el dibujo de las gráficas estadísticas (Androidplot.jar).
- **Res:** contiene todos los recursos que utiliza la aplicación. Estos recursos generalmente son definiciones de pantallas, imágenes estáticas, ficheros de audio así como recursos necesarios para la internacionalización de las aplicaciones. El directorio res se divide a su vez en:
  - **Drawable:** compuesto por elementos gráficos de la aplicación como imágenes estáticas y ficheros xml que definen estilos aplicables a los objetos. En nuestro caso, en la carpeta drawable están todos los iconos de la aplicación, así como un conjunto de ficheros que nos han permitido personalizar los elementos de la interfaz gráfica: bordes redondeados, degradados, fondos, etc.  
 La plataforma Android permite que existan varias carpetas de recursos gráficos con el objeto de hacer aplicaciones adaptadas a diferentes terminales. De esta forma, existen las carpetas drawable-hdpi, drawable-ldpi y drawable-mdpi que pueden contener elementos adicionales (o sobrescribir a la carpeta drawable). Si estas carpetas tienen elementos, el terminal utilizará la carpeta de recursos en función de la densidad de píxeles que tenga su pantalla. Así, se podría definir una imagen de fondo con más resolución dentro de la carpeta drawable-hdpi y si se ejecuta la aplicación en una tablet, esta cargará la imagen más pesada; la imagen con menor resolución para aquellos dispositivos con pantallas más pequeñas.
  - **Layout:** contiene los XMLs que definen las diferentes pantallas que componen la aplicación.
  - **Values:** formada por un único fichero XML que contiene las diferentes cadenas de texto utilizadas. El objetivo de separar las cadenas del texto del código o los xmls de definición de vistas, es la internacionalización de la aplicación.  
  
 De esta forma, para adaptar la aplicación a otro idioma basta con crear un nuevo fichero (strings.xml) con las cadenas traducidas y colocarlas dentro de una nueva carpeta values-XX.
- **Src:** Contiene el código de la aplicación. Los ficheros class resultantes de la compilación del código se almacenarán en un directorio bin.

- **AndroidManifest.xml:** este fichero cuelga de la raíz del proyecto, es obligatorio en todas las aplicaciones Android y se encarga de definir algunos parámetros de la aplicación.

Dentro de la carpeta **src**, en nuestro proyecto se ha optado por dividir el código en capas, donde cada paquete se encarga de una parte concreta de la aplicación:

- **paquete com.quique.pfc.Activities** - contiene todas las Activities de la aplicación, es decir, aquellas clases encargadas de definir el comportamiento de las pantallas así como de establecer qué XML define su interfaz. Las Activities implementadas no contienen ninguna lógica de negocio, simplemente instancian los diferentes elementos de la interfaz (botones, formularios, listas) y se encarga de establecer el comportamiento de cada elemento (invocando al correspondiente servicio de la lógica de negocio).
- **paquete com.quique.pfc.datos** - representa la capa de acceso a datos. La aplicación cuenta con dos fuentes de datos diferentes, por un lado, la propia base de datos de la aplicación donde se almacenan las cuentas de usuario y se registra la actividad de cada uno de ellos, y por otro, el servidor de Moodle. Para cada una de las fuentes de datos se han implementado un DAO (Data Access Object) que tiene los diferentes métodos de acceso/borrado/modificación para cada fuente de datos. El objetivo de crear los diferentes DAOs en vez de llamar directamente a las consultas de datos en la lógica de negocio es hacer el sistema más robusto y fácilmente modificable ante cambios en el origen de los datos.

Por ejemplo, si en un futuro cambian todos los servicios web de Moodle o se quieren almacenar las cuentas de usuario en un servidor externo, únicamente tendríamos que modificar el DAO correspondiente sin modificar su interfaz, y la lógica de negocio y la presentación serían igualmente válidas y el cambio para estas capas sería transparente.

- **paquete com.quique.pfc.dtos** - Contiene diferentes POJOs (Plain Old Java Object) que modelan los datos que viajan entre las diferentes capas de la aplicación. Para cada tipo de objeto presente en la lógica de negocio (cursos, usuarios, participantes, contenido...) se ha creado un DTO (Data Transfer Object) con todos los atributos públicos.
- **paquete com.quique.pfc.servicios** – formado por las clases que implementan la lógica de negocio de la aplicación. Estos servicios de negocio serán invocados desde

las clases que únicamente se encargan de la presentación de los datos (Activities) y se nutren de los datos proporcionados por los DAOs. El objetivo de la implementación de estas clases es definir claramente cuál es la funcionalidad de la aplicación y separar la presentación del acceso a datos. Adicionalmente, la clase de negocio encargada de la interacción con el servidor Moodle se encarga de tener cacheados cierto tiempo los datos con el objetivo de mejorar el rendimiento de la aplicación y minimizar el tráfico de red.

- **paquete com.quique.pfc.utils** - paquete en el cual se alojan un conjunto de clases estáticas con diferentes métodos que son utilizados en el resto de capas de la aplicación: utilidades para el manejo de cadenas, peticiones http, cálculo de listas, etc. Todas estas clases son estáticas porque únicamente ofrecen funcionalidad y no tiene sentido instanciarlas ya que no tienen ningún atributo ni estado que mantener.

## 6.2 Diseño de la capa de gestión de datos

La aplicación mantiene su propia base de datos para el almacenamiento de las cuentas de usuario que han sido utilizadas para hacer login y para el registro de la actividad. Esta base de datos está implementada, al igual que todas las aplicaciones Android, en el SGBD SQLite.

### 6.2.1 Diseño conceptual

- Tipos de entidades:

En la tabla 6.1 se muestran las entidades que intervienen, sus atributos y el tipo de entidad.

Nombre	Atributos	Tipo
Cuentas	idCuenta, url, usuario, password, imagen	Fuerte
Actividad	idActividad, idCuenta, actividad, fecha	Fuerte

Tabla 6.1: Tabla de entidades

- Atributos de las entidades:

En las tablas 6.2 y 6.3 se describen los atributos de cada una de las entidades.

Nombre	Descripción	Tipo	Nulo
idCuenta	Identificador único de cada fila. Campo generado y autoincrementado por el SGBD SQLite. De esta forma, al introducir datos sobre la tabla, no tenemos que especificar este valor ya que es el propio SQLite quien se encarga de insertarlo.	Clave primaria	No
url	Dirección web del servidor de Moodle al que pertenece la cuenta. Debe incluir el prefijo http o https (la aplicación funciona con los dos protocolos..	Simple	No
usuario	Nombre de usuario en Moodle	Simple	No
password	<p>Contraseña en Moodle. El servidor de Moodle exige que las contraseñas tengan al menos un carácter en mayúsculas y un carácter especial (numérico, ;, _, etc).</p> <p>Para almacenarlo en la BBDD de nuestra aplicación, se ha codificado dicha contraseña utilizando el algoritmo AES. Se detalla este proceso en el punto 7.6 de la memoria.</p> <p>Sin embargo, para llamar al servicio web de autenticación se le debe mandar el password sin cifrar.</p>	Simple	No

Tabla 6.2: Tabla de atributos de la entidad Cuentas

Nombre	Descripción	Tipo	Nulo
idActividad	Identificador único de cada fila. Campo generado y autoincrementado por el SGBD SQLite. De esta forma, al introducir datos sobre la tabla, no tenemos que especificar este valor ya que es el propio SQLite quien se encarga de insertarlo.	Clave primaria	No
idCuenta	Identificador único de la cuenta de usuario en la base de datos alojada en nuestro terminal.	Simple	No
actividad	Se identifica el tipo de actividad a registrar. En nuestro caso, se ha utilizado un 1 cuando se trate de login, un 2 cuando se trate de una subida de fichero y un 3 cuando se consulte el detalle de un curso.	Simple	No



fecha	Fecha y hora en la que se produce la actividad. Será de gran utilidad cuando se saquen las gráficas estadísticas.	Simple	No
-------	---	--------	----

Tabla 6.3: Tabla de atributos de la entidad Actividad

- Relaciones:

Las relaciones que intervienen son:

- **Realiza** Relación que existe entre las cuentas y las actividades. Se crea un atributo de la relación para determinar la fecha de cuando se produce y el tipo.
- **Conexiones** Relación que existe entre los usuarios con las conexiones al sistema.

- Diagrama de entidad-relación:

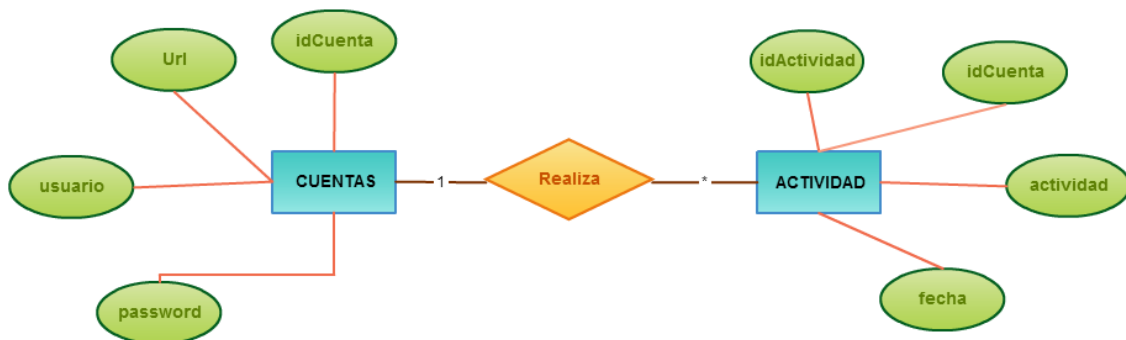


Figura 6.2: Esquema Entidad-Relación

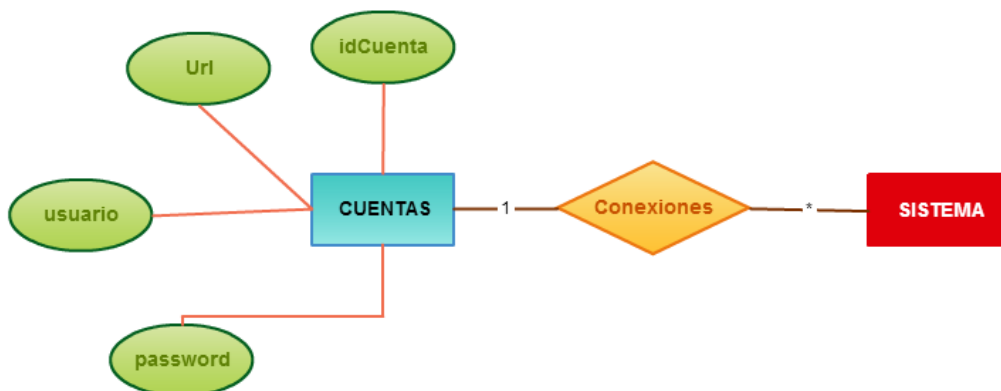


Figura 6.3: Esquema Entidad-Relación – Conexiones al sistema

### 6.2.2 Diseño lógico

Una vez obtenidas las tablas, se va a proceder a aplicar el proceso de normalización. La normalización, en el contexto de las base de datos, son una serie de reglas que permiten a los diseñadores de bases de datos desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede.

Además, una base de datos normalizada es más fácil de entender, y lo que es más importante, tiene un consumo de espacio menor debido a que se reduce la repetición de datos.

En nuestra base de datos se van a aplicar 4 grados de normalización, los cuales son:

- **1ª Forma Normal:** establece que todos los atributos de la tabla deben ser atómicos (indivisibles), existe una clave primaria con atributos y estos son no nulos. Con esto se consiguen eliminar los valores repetidos en una base de datos.
- **2ª Forma Normal:** establece que aquellos datos que no dependen de la clave primaria se deben eliminar y separar dentro de sus propias tablas.
- **3ª Forma Normal:** una tabla esta en 3FN si todos los datos son dependientes funcionalmente de la clave primaria y no existe dependencias transitivas, es decir, que un columna dependa de otra y ninguna de ellas sea clave primaria.
- **Forma Normal de Boyce-Codd (o FNBC):** es una 3FN ligeramente más fuerte. Una tabla se encuentra en “FNBC” si está en 3FN y si sólo existen dependencias funcionales elementales que dependan de la clave primaria o de cualquier clave alternativa.

Ahora aplicaremos los 4 grados de normalización a nuestras tablas:

- **1ª Forma Normal:** podemos considerar que todas las tablas descritas anteriormente están en 1FN ya que todos sus atributos son atómicos, contienen una única clave primaria y estas no son nulas.
- **2ª Forma Normal:** todas las tablas se encuentran directamente en 2FN debido a que están en 1FN y su clave primaria está compuesta por un solo atributo.
- **3ª Forma Normal:** todas las tablas se encuentran el 3FN ya que todas se encuentran en 2FN y ningún atributo que no sea clave primaria de la tabla es dependiente transitivamente de una clave candidata.
- **FNBC:** en este caso, todas la tablas estarán en FNBC debido a que están en 3FN y no existen dependencias no triviales a una parte de la clave primaria.

### 6.2.3 Diseño físico

Cuando el usuario accede por primera vez a la aplicación y realiza el primer login contra un sitio Moodle, se lleva a cabo la creación de la base de datos y las estructuras de datos necesarias.

A continuación mostramos las secuencias para la creación de la base de datos y las estructuras utilizadas:

```

61  /** Objeto para realizar las operaciones contra la BBDD SQLite*/
62  private DatabaseHelper mDbHelper;
63  /** Contexto de la aplicacion */
64  private final Context mContext;
65  /** Instancia de la bbdd de la aplicacion */
66  private SQLiteDatabase mDb;
67
68  /** Constante para definir el nombre de la base de datos*/
69  private static final String DATABASE_NAME = "moodleClientDatabase.db";
70  /** Constante para definir el nombre de la tabla de cuentas de usuario */
71  private static final String DATABASE_CUENTA = "cuentas";
72  /** Constante para definir el nombre de la tabla de actividad*/
73  private static final String DATABASE_ACTIVIDAD = "actividad";
74  /** Version de la BBDD */
75  /**
76   *Ante un cambio de estructura en la BBDD hay que incrementar la
77   *version para que en la inicializacion de la aplicacion vuelva a
78   *ejecutar los scripts de borrado y creacion de la BBDD
79   */
80  private static final int DATABASE_VERSION = 8;
81
82  /**SQLs para la cracion de la BBDD*/
83  /**SQL de creacion de la tabla de cuentas de usuario*/
84  public static final String DATABASE_CREAR_CUENTAS =
85      "create table cuentas (idCuenta integer primary key autoincrement, " +
86      "url text not null, " +
87      "usuario text not null, password text not null, imagen text );";
88
89
90  /**SQL de creacion de la tabla de actividad*/
91  public static final String DATABASE_CREAR_ACTIVIDAD =
92      "create table actividad (idActividad integer primary key autoincrement, "+
93      "idCuenta integer not null, " +
94      "actividad integer not null, " +
95      "fecha date not null);";
96
97  /** El constructor del adaptador de base de datos
98   * necesita el contexto de la aplicacion */
99  public BDAdapter(Context ctx) {
100      this.mContext = ctx;
101  }
102  /** Abre la conexion con la base de datos.
103   * Si no existe la BBDD, la crea */
104  public BDAdapter open() throws SQLException {
105      mDbHelper = new DatabaseHelper(mContext);
106      mDb = mDbHelper.getWritableDatabase();
107      return this;
108  }

```

```

109      /** Cierra la conexión con la base de datos */
110      public void close() {
111          mdbHelper.close();
112      }
113
114
115      /** Clase estática con utilidades de la base de datos. Se encargará de ejecutar
116       * el 'script' de creación de las tablas */
117      private static class DatabaseHelper extends SQLiteOpenHelper
118      {
119          DatabaseHelper(Context context)
120          {
121              super(context, DATABASE_NAME, null, DATABASE_VERSION);
122          }
123          /** Este método se ejecuta cuando se crea la BBDD por primera vez */
124          @Override
125          public void onCreate(SQLiteDatabase db)
126          {
127              Log.i("BBDD", "Se crea la BBDD de la aplicación");
128              db.execSQL(DATABASE_CREAR_CUENTAS);
129              db.execSQL(DATABASE_CREAR_ACTIVIDAD);
130          }
131          /** Este método se ejecuta ante una actualización de la BBDD */
132          @Override
133          public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
134          {
135              Log.i("BBDD", "Actualización de la BBDD, se borran las tablas.");
136              //Borra las tablas existentes...
137              db.execSQL("DROP TABLE IF EXISTS " + DATABASE_ACTIVIDAD);
138              db.execSQL("DROP TABLE IF EXISTS " + DATABASE_CUENTA);
139              Log.i("BBDD", "Tablas borradas correctamente");
140              //.. y las vuelve a crear
141              onCreate(db);
142          }
143      }
144  }
145

```

Figura 6.4: Secuencias para la creación de la base de datos

## 6.3 Diseño de la capa de dominio

### 6.3.1 Diagrama de clases de diseño

Para una mejor comprensión y lectura de los diagramas de clases, se ha optado por dividirlos según el paquete principal al que corresponden en la aplicación.

- Diagrama de clases del paquete Activities:

A continuación se muestra el diagrama de clases de diseño. En el diagrama aparecerán los métodos de mayor relevancia.

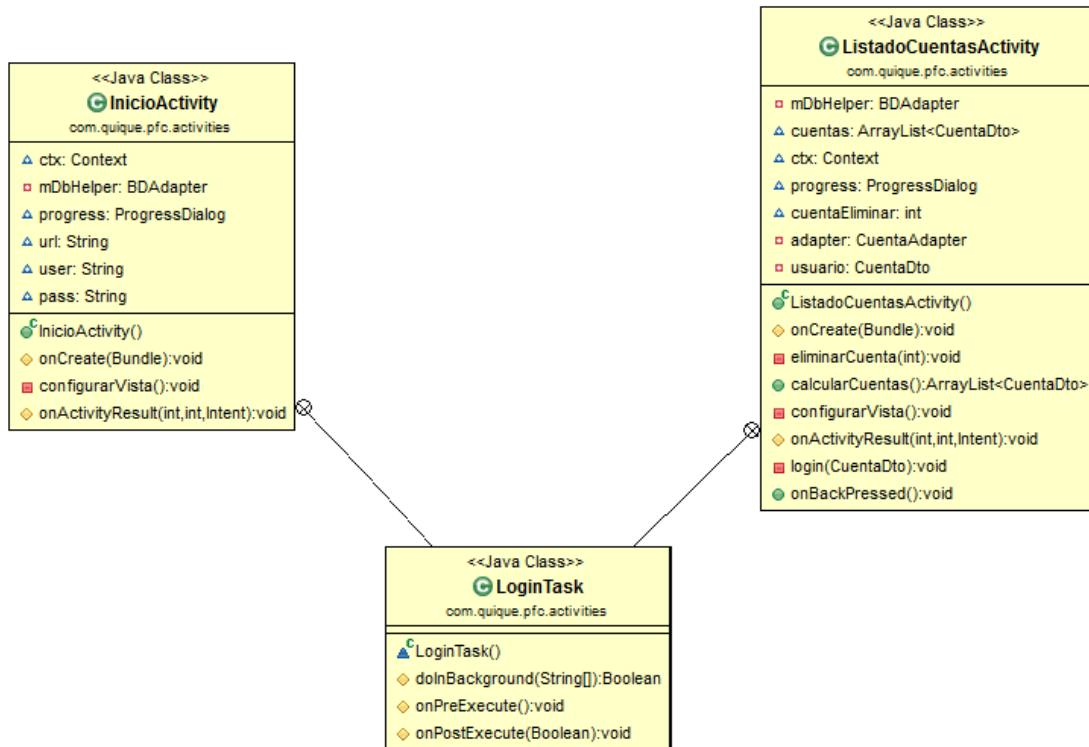


Figura 6.5: Diagrama de clases de diseño – Login

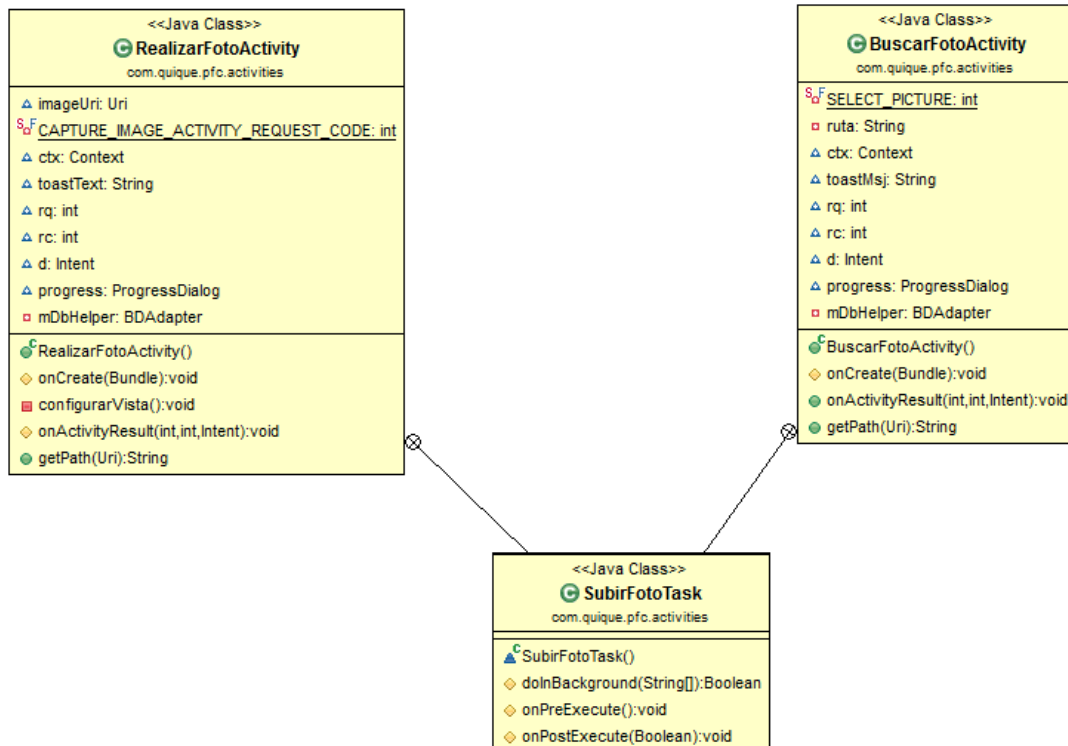


Figura 6.6: Diagrama de clases de diseño – Subida Fotos

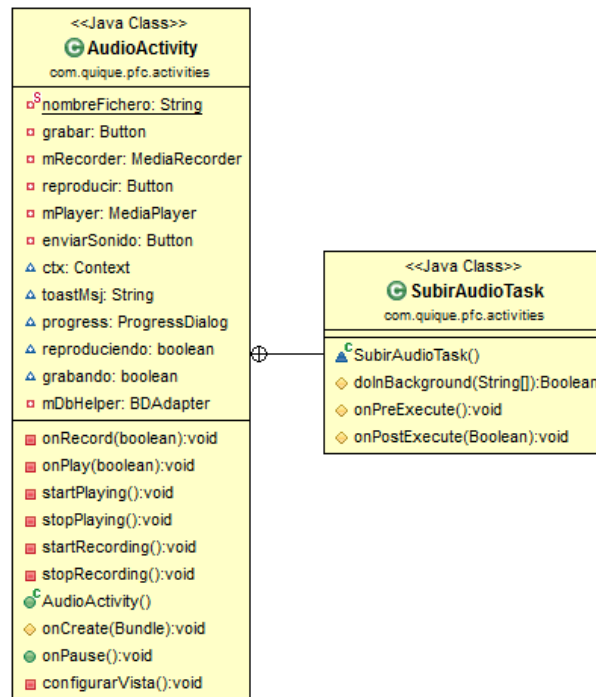


Figura 6.7: Diagrama de clases de diseño – Subida Audios

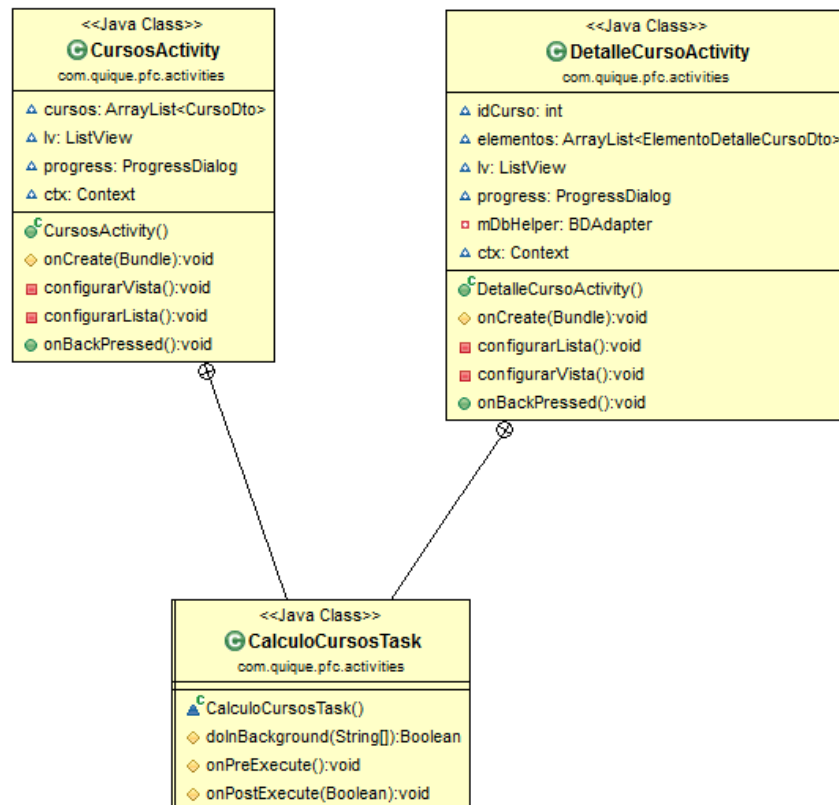


Figura 6.8: Diagrama de clases de diseño – Cursos

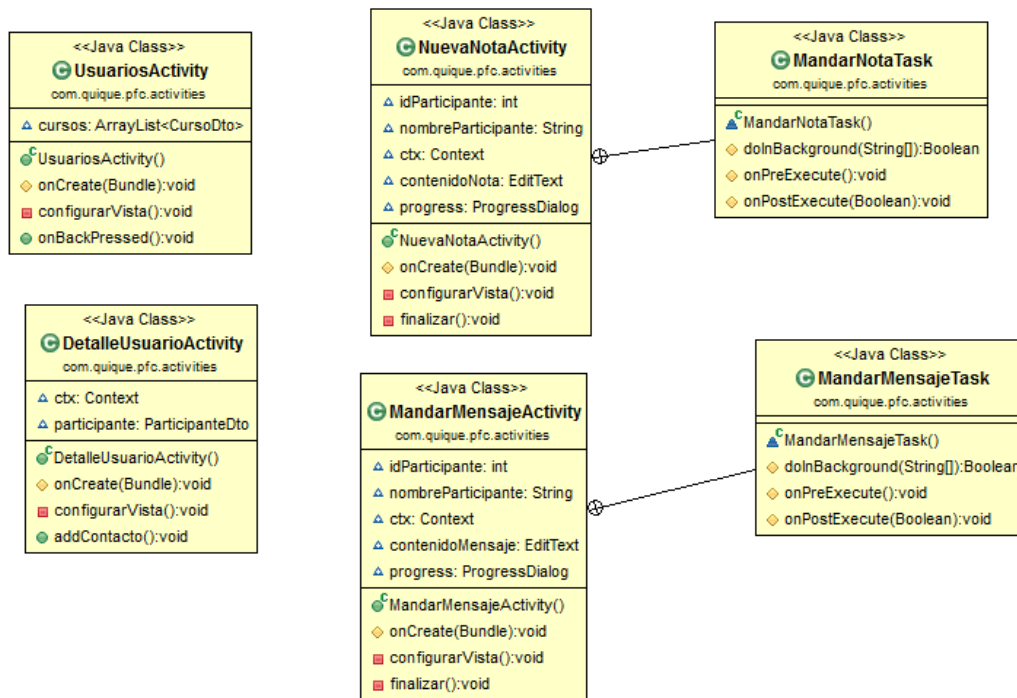


Figura 6.9: Diagrama de clases de diseño – Participantes

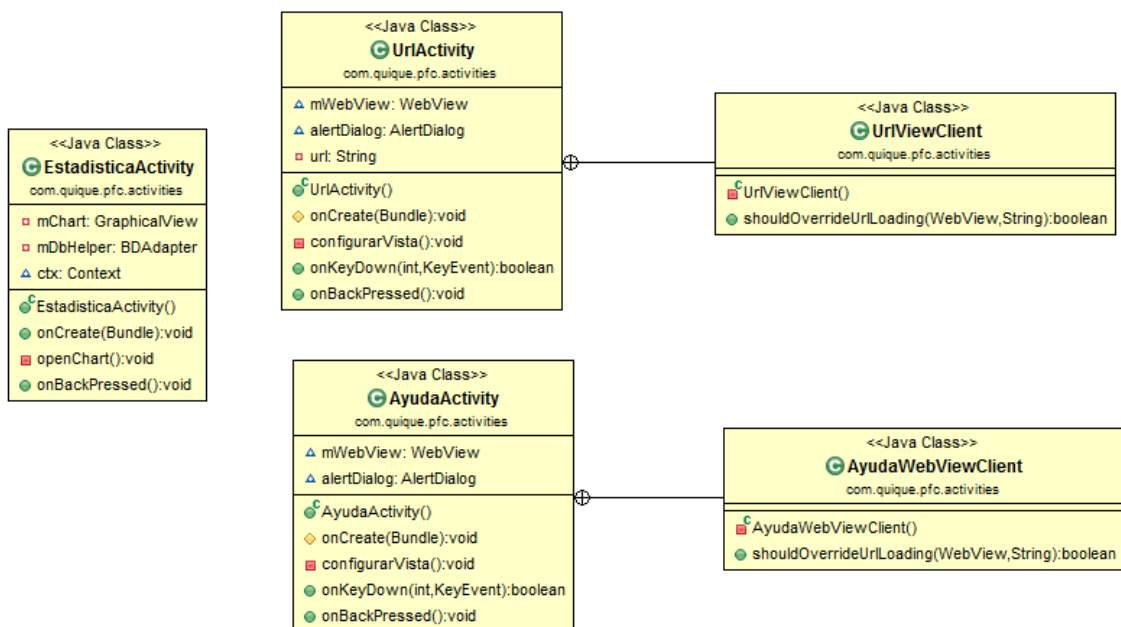


Figura 6.10: Diagrama de clases de diseño – Estadísticas, ayuda y acceso Web

- Diagrama de clases del paquete Datos:

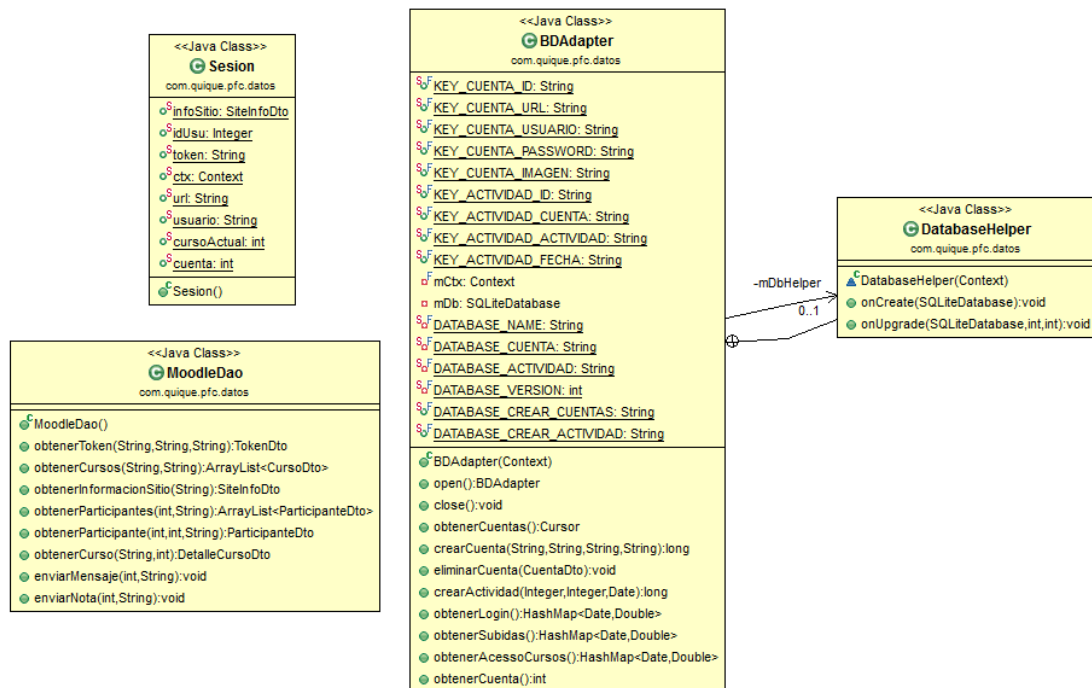


Figura 6.11: Diagrama de clases de diseño – Acceso a datos

- Diagrama de clases del paquete Dtos:

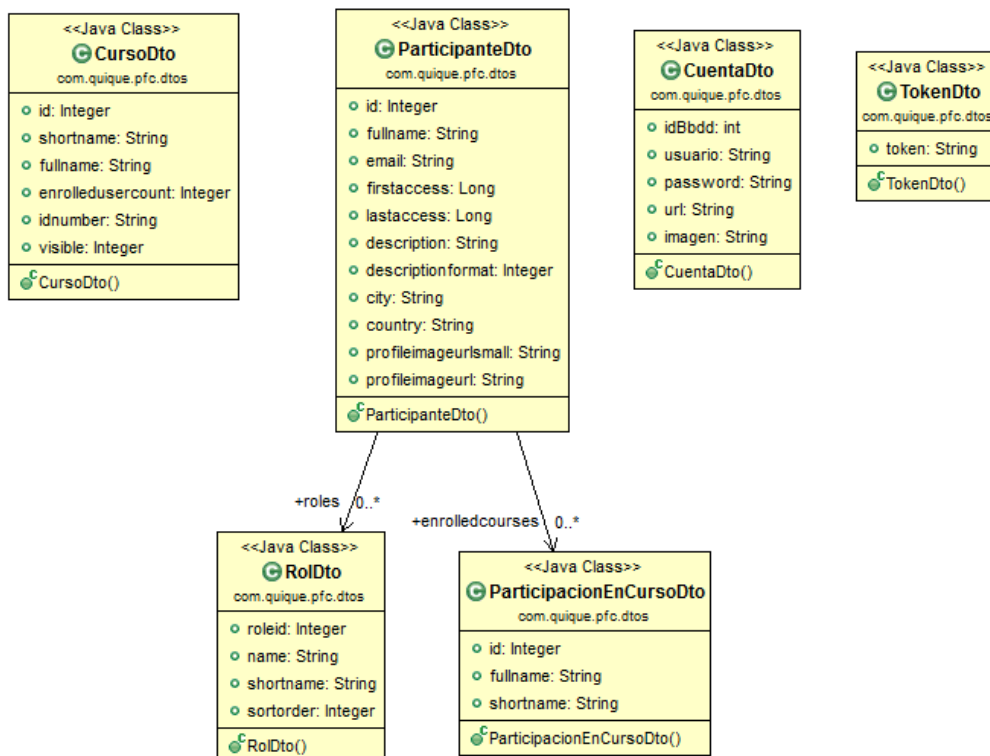


Figura 6.12: Diagrama de clases de diseño – DTOs (parte I)



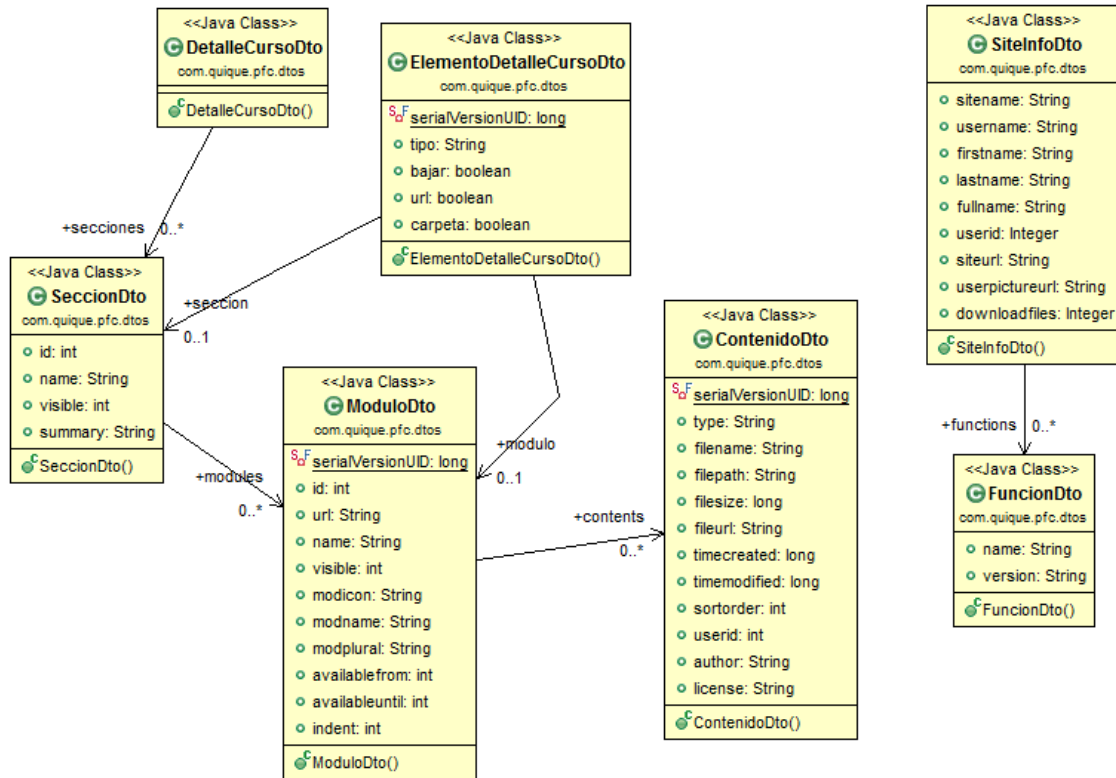


Figura 6.13: Diagrama de clases de diseño – DTOs (parte II)

- Diagrama de clases del paquete Servicios:

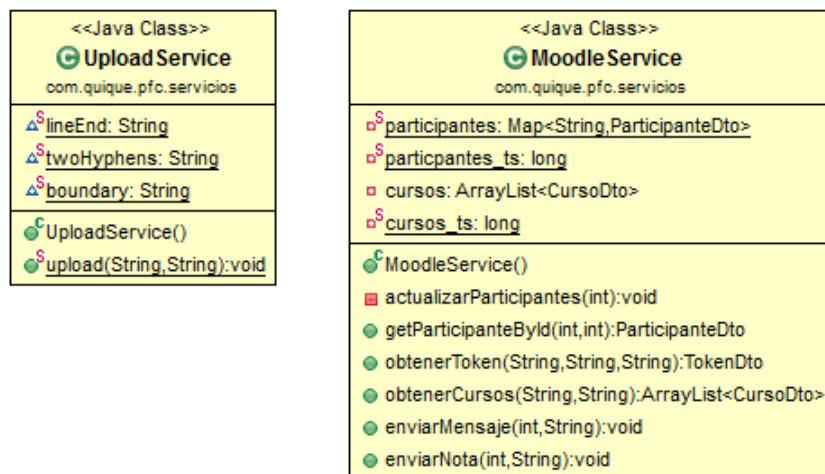


Figura 6.14: Diagrama de clases de diseño – Servicios

- Diagrama de clases del paquete Utils:

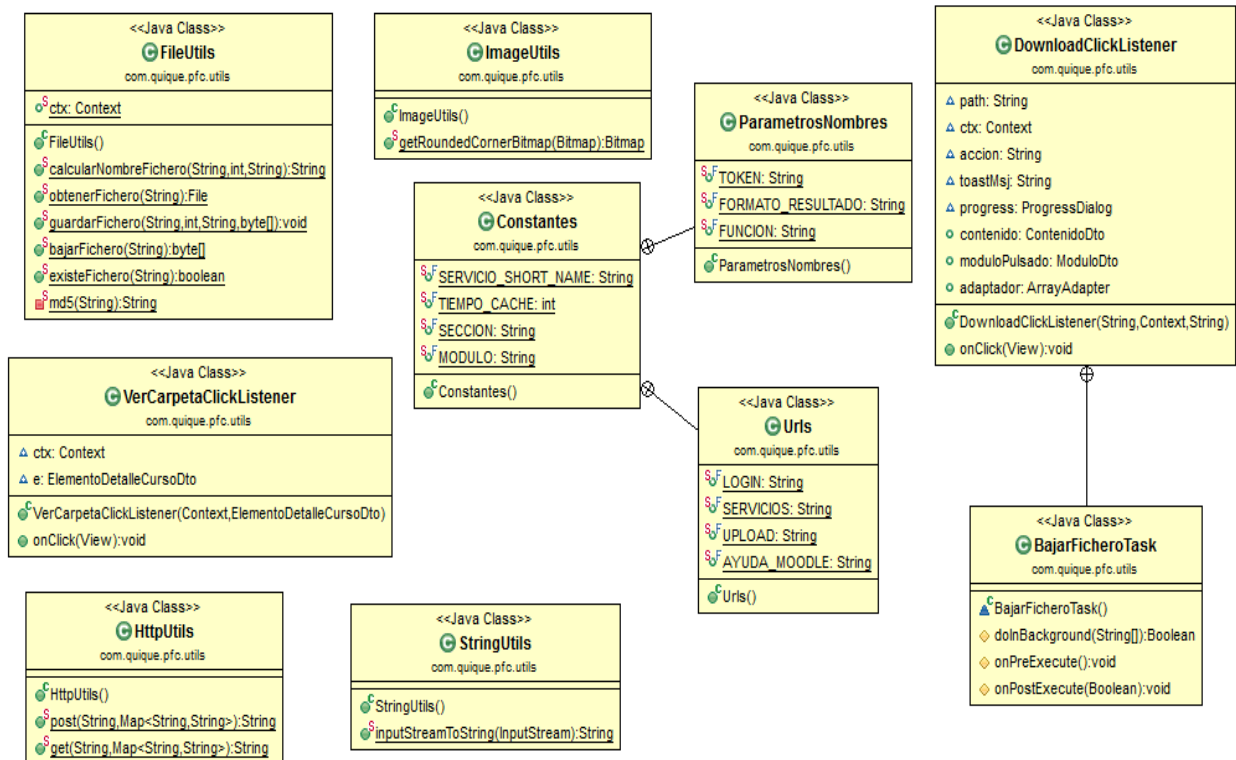


Figura 6.15: Diagrama de clases de diseño – Utils

### 6.3.2 Diagrama de secuencia

En este apartado se irán presentando los diferentes diagramas de secuencia, junto con la explicación general de lo que ellos representan.

Al abrir la aplicación, se mostrará una pantalla para conectarnos al sitio Moodle deseado con nuestras credenciales. Sin pasar estas comprobaciones, no podremos acceder al resto de funcionalidades de la aplicación. En la figura 7.19 podemos observar el diagrama de la acción descrita con anterioridad.

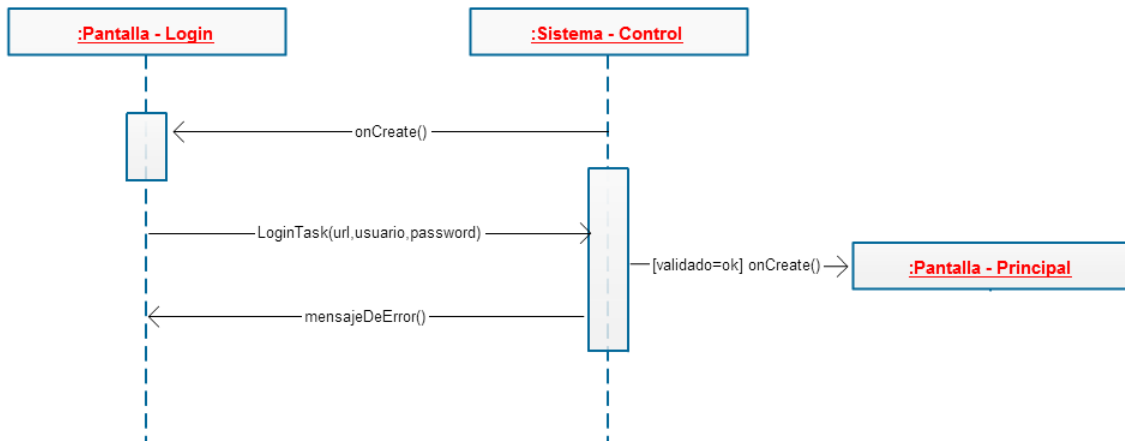


Figura 6.16: Diagrama de interacción – Login

Además, como se ha comentado anteriormente, es posible acceder a través de una lista de cuentas almacenadas con las que ya hemos login previamente. Para ello, se accederá a la pantalla de Listado de Cuentas desde la pantalla de Login y se seleccionará una de las cuentas disponibles. Podemos ver cómo sería la secuencia en la figura 7.20

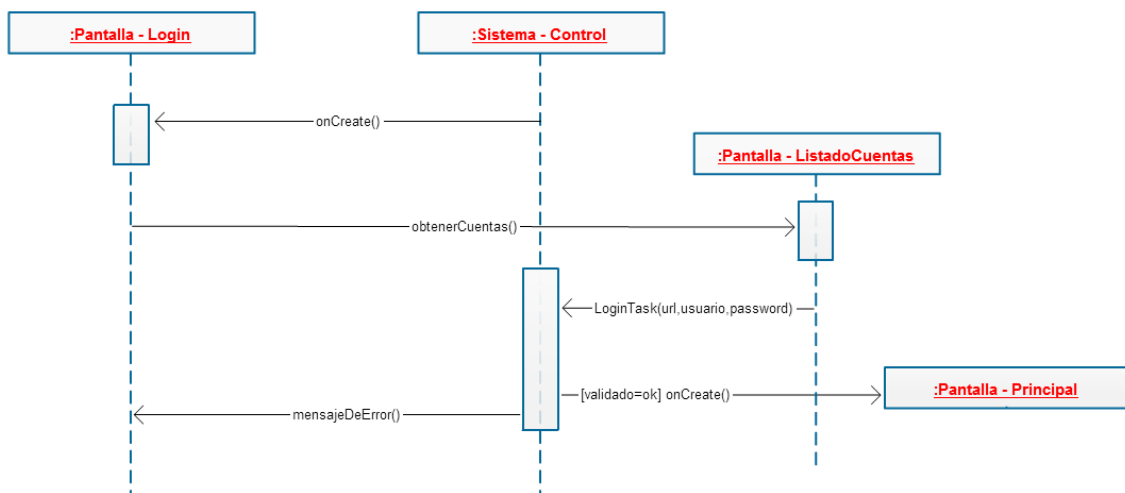


Figura 6.17: Diagrama de interacción – Login desde cuenta almacenada

Desde la pantalla principal se mostrará un menú en el que se agrupan las opciones que tenemos en la aplicación. Así pues, en la figura 7.21 pueden observarse todas las operaciones que podemos ejecutar desde el menú principal.

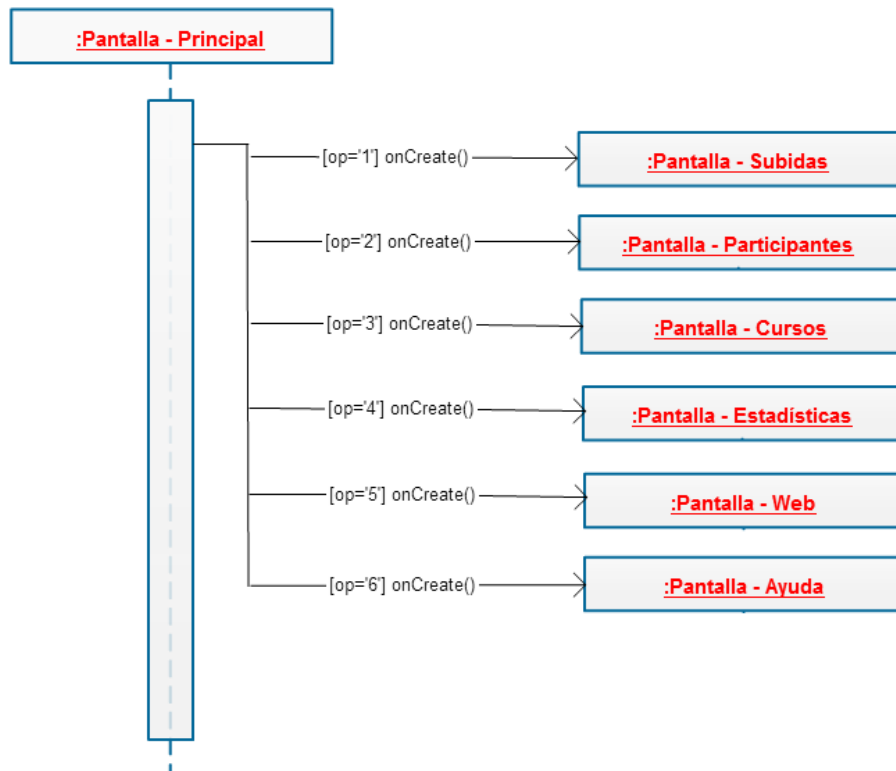


Figura 6.18: Diagrama de interacción – Principal

En el siguiente diagrama se representa el caso de uso en el que el usuario selecciona una foto ya existente en el terminal y la sube a su sitio privado.

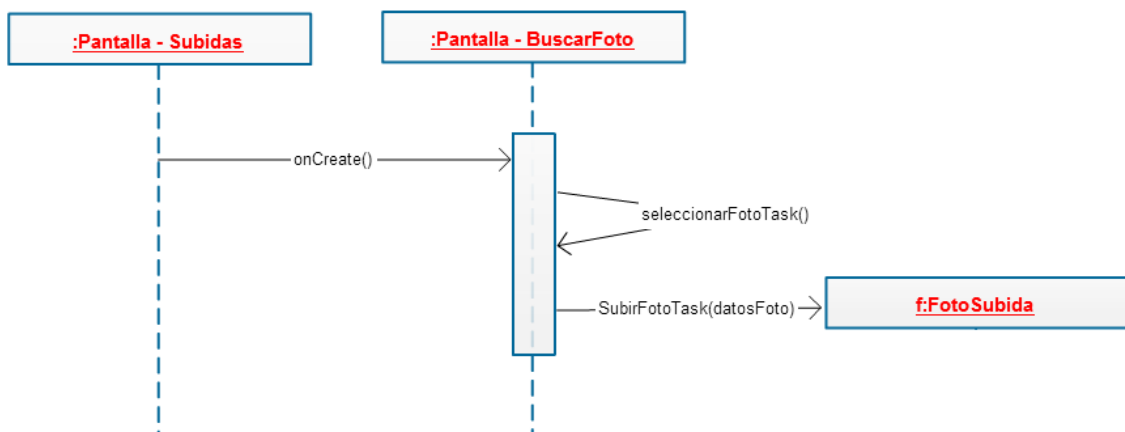


Figura 6.19: Diagrama de interacción – Subir foto almacenada

Para el caso en el que el usuario realiza la foto y posteriormente la sube, la secuencia aparece representada en la figura 7.23. Este caso es similar al de realizar un vídeo y subirlo, por ello obviamos su representación.

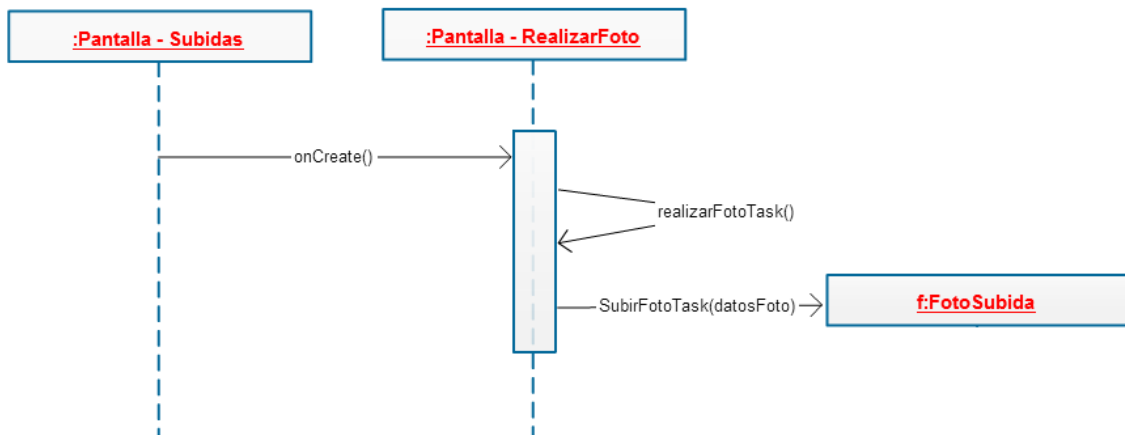


Figura 6.20: Diagrama de interacción – Realizar foto y subirla

La última acción posible a realizar desde el menú de Subidas es la de grabar un fichero de audio y subirlo al espacio privado. La secuencia para realizar dicha acción es la siguiente:

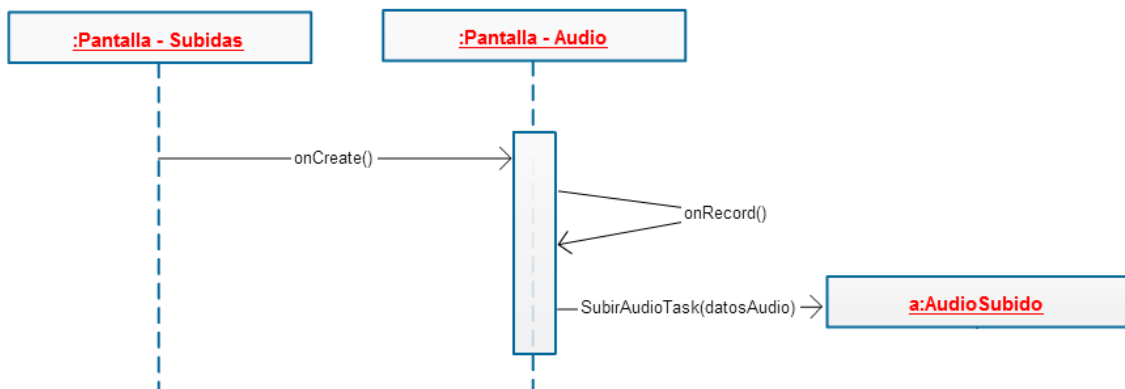


Figura 6.21: Diagrama de interacción – Grabar audio y subirlo

Los próximos diagramas van relacionados con los participantes pertenecientes a cada uno de los cursos. Podremos listar dichos participantes, ver información detallada de cada uno de ellos y mandar mensajes. Así, en la figura 7.24 se representa la forma de listar los participantes. En la 7.25 podemos ver cómo alcanzar la pantalla de detalle de un usuario seleccionado. Por último, en la 7.26 se representa la acción de mandar un mensaje a uno de los participantes.

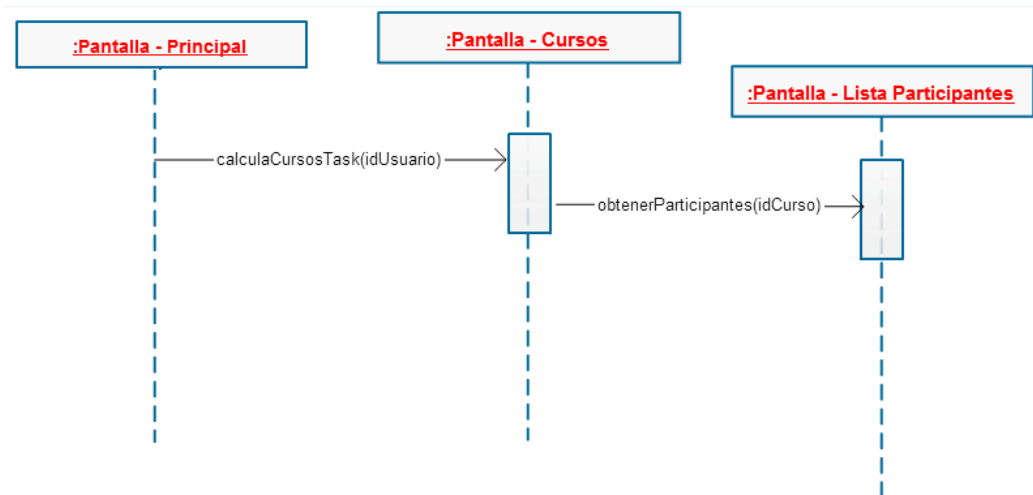


Figura 6.22: Diagrama de interacción – Listar participantes

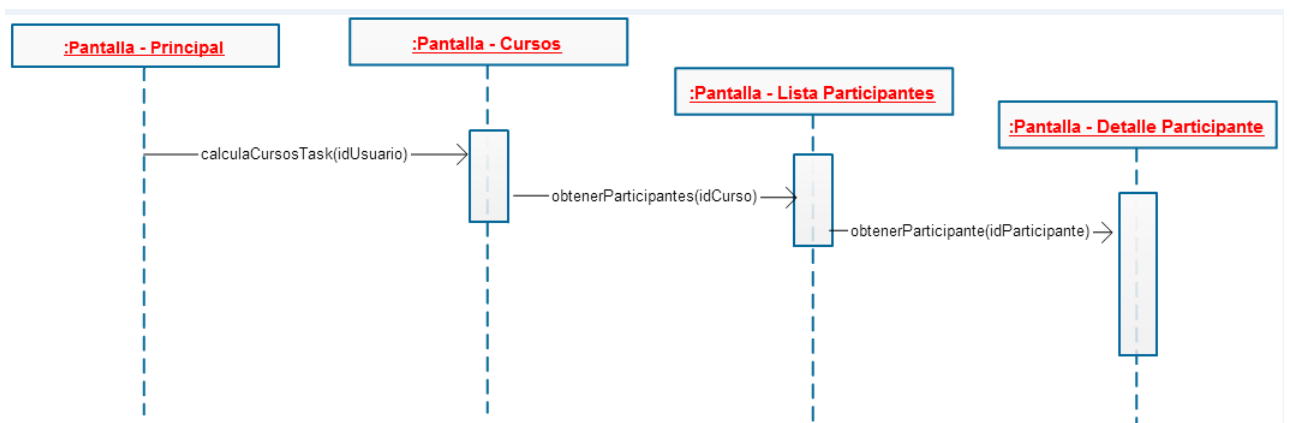


Figura 6.23: Diagrama de interacción – Detalle de un participante

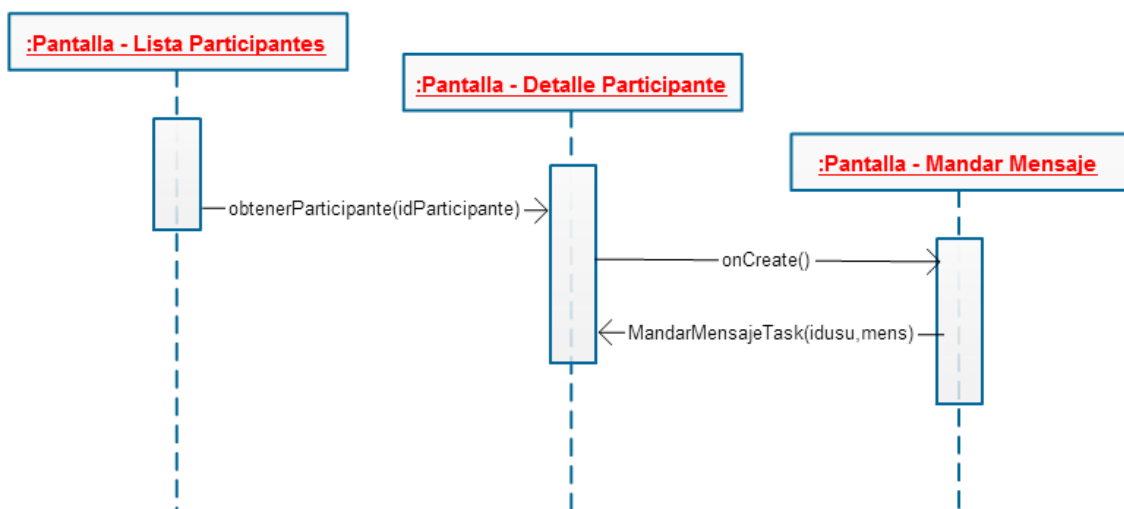


Figura 6.24: Diagrama de interacción – Mandar mensaje a un participante

Pasamos a tratar ahora los casos relacionados con los cursos. La primera acción que representaremos será la de listar los cursos que tiene asignado el usuario.

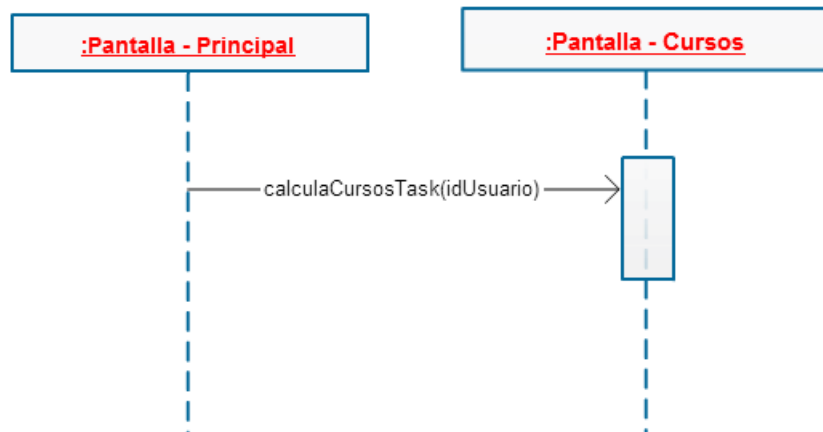


Figura 6.25: Diagrama de interacción – Cursos asignados al usuario

Una vez se encuentre listada la lista de cursos, podemos acceder al contenido de uno de ellos. Este caso aparece reflejado en la figura 7.28

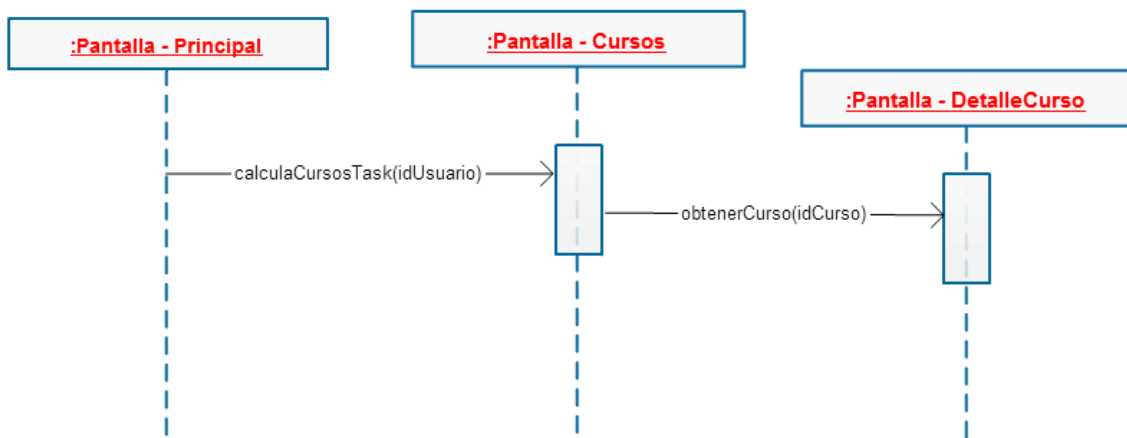


Figura 6.26: Diagrama de interacción – Detalle de un curso

Una vez que se listan los elementos de un curso, el usuario tendrá la opción de descargar y ver los contenidos, así como de listar el contenido de una carpeta. En los 2 próximos diagramas aparecen representadas ambas acciones.

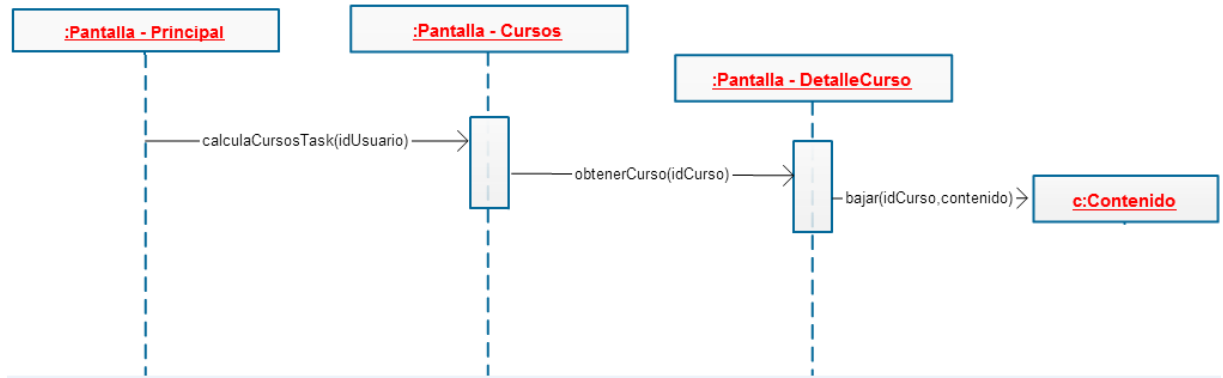


Figura 6.27: Diagrama de interacción – Descargar contenido

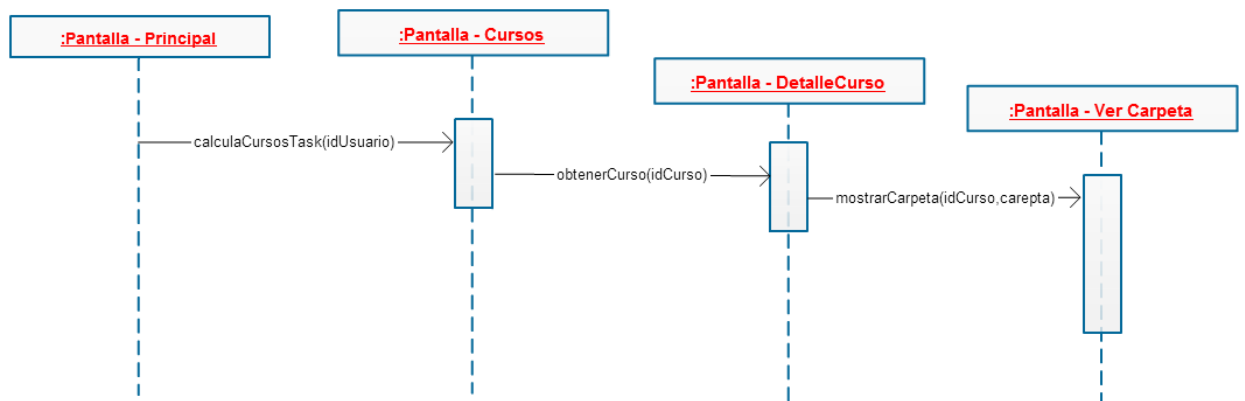


Figura 6.28: Diagrama de interacción – Visualizar contenido carpeta

Otro de los casos de uso sería el correspondiente a dibujar las estadísticas de un usuario desde el terminal en el que se encuentra la aplicación. Se mostrarán datos de login, subidas y acceso a cursos.

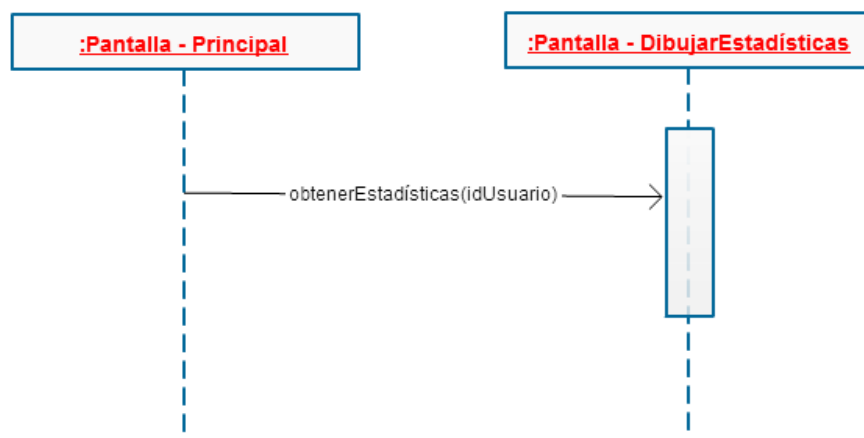


Figura 6.29: Diagrama de interacción – Estadísticas



El próximo diagrama de interacción refleja el caso de acceso a la versión web del sitio Moodle al cual nos encontramos conectados.

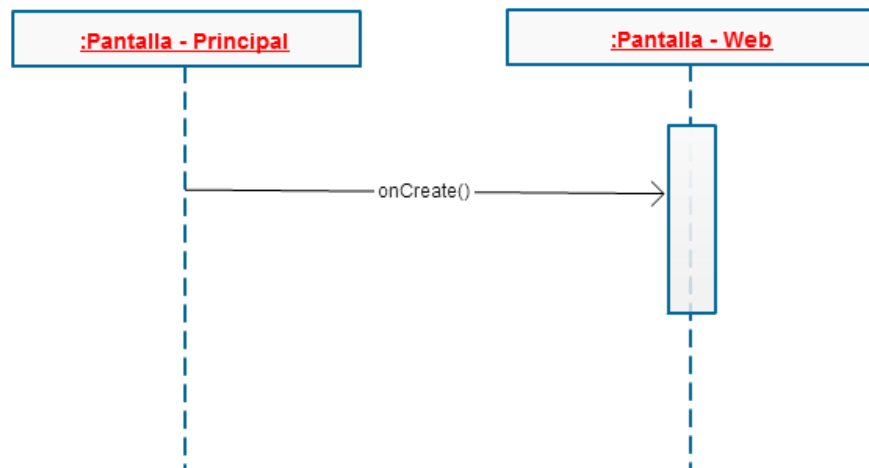


Figura 6.30: Diagrama de interacción – Versión web

Finalmente, se representa el caso de uso en cual se accede a la ayuda oficial de Moodle.

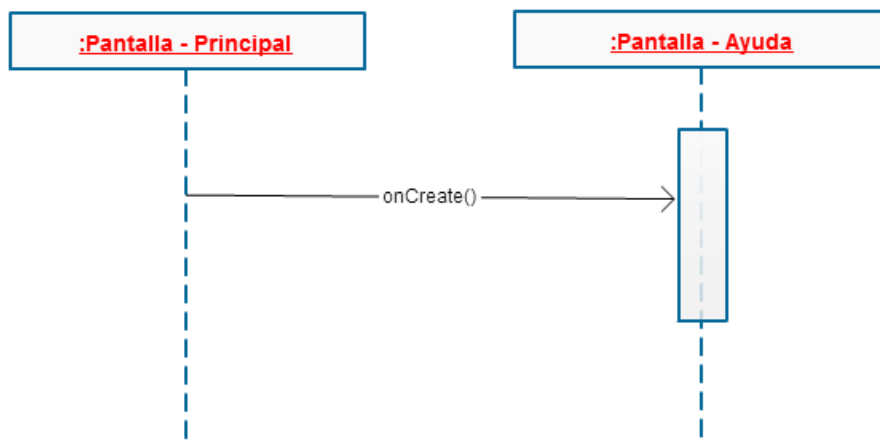


Figura 6.31: Diagrama de interacción – Ayuda Moodle

### 6.3.3 Contratos de las operaciones

En este punto se van a describir los contratos de las operaciones vistas en los diagramas de secuencia reflejados en el punto anterior.

- Pantalla Login
  - *onCreate()* (constructor): lleva a cabo la creación de la pantalla de login para que el usuario introduzca la url del sitio Moodle al que se quiere conectar, su nombre de usuario y el password.
  - *LoginTask(url, usuario, password)*: credenciales de acceso del usuario para la aplicación.
  - *mensajeDeError()*: mensaje del sistema para el usuario en caso de que las credenciales de accesos sean incorrectas.
  - *onCreate()* (constructor): se encarga de crear la pantalla principal de la aplicación.
  - *obtenerCuentas()*: consulta para obtener todas las cuentas de usuario con las que ya se ha accedido desde el terminal donde el aplicativo se encuentra instalado.
- Menú Principal
  - *onCreate()* (constructor): creará la pantalla según la opción del menú elegida (subidas, participantes, cursos, estadísticas, web y ayuda)
- Pantalla Subidas
  - *onCreate()* (constructor): se muestra la galería de imágenes almacenadas en el terminal.
  - *seleccionarFotoTask()*: el usuario selecciona la imagen que desea subir a su espacio privado.
  - *SubirFotoTask(datosFoto)*: se sube la imagen seleccionada al espacio privado del usuario.
  - *onCreate()* (constructor): se muestra la cámara del terminal para que el usuario realice una foto o vídeo.
  - *RealizarFotoTask()*: el usuario realiza la foto que desee para posteriormente subirla.
  - *onCreate()*(constructor): creación de la pantalla de audio donde el usuario podrá grabar un sonido.

- *onRecord()*: grabación de audio por parte del usuario.
  - *SubirAudioTask(datosAudio)*: se sube el audio previamente grabado al espacio privado del usuario.
- Pantalla Participantes
    - *calculaCursosTask(idUsuario)*: se obtiene la lista de cursos que tiene asignados el usuario en el sitio Moodle al que se encuentra conectado.
    - *obtenerParticipantes(idCurso)*: se obtiene la lista de participantes para el curso seleccionado.
    - *obtenerParticipante(idParticipante)*: acceso a la información detallada del participante elegido por el usuario.
    - *onCreate()(constructor)*: se muestra la pantalla para escribir el mensaje que posteriormente podremos enviar al participante.
    - *mandarMensajeTask(idParticipante, mensaje)*: acción a través de la cual se envía el mensaje escrito por el usuario al participante seleccionado.
- Pantalla Cursos
    - *calculaCursosTask(idUsuario)*: se obtiene la lista de cursos que tiene asignados el usuario en el sitio Moodle al que se encuentra conectado.
    - *obtenerCurso(idCurso)*: acceso al detalle del curso seleccionado. Se muestran sus distintas secciones y contenidos.
    - *bajar(idCurso, contenido)*: descarga al terminal del contenido seleccionado. Si el terminal tiene el visor apropiado para ver el contenido descargado, el usuario tendrá además la opción de visualizar el contenido.
    - *mostrarCarpeta(idCurso, carpeta)*: se muestran los contenidos incluidos dentro de la carpeta seleccionada.
- Estadísticas
    - *obtenerEstadística(idUsuario)*: se dibujan los datos estadísticas del usuario para el sitio Moodle al que se encuentra conectado (sólo los datos para el terminal actual).
- Web
    - *onCreate()(constructor)*: se muestra la versión web del sitio Moodle al que el usuario se encuentra conectado.

- Ayuda
  - `onCreate() (constructor)`: se muestra la ayuda oficial de Moodle.

## 6.4 Diseño de la capa de presentación

Llegados a este punto, mostraremos el diseño de la capa de la presentación. Tal y como se ha comentado en puntos anteriores, esta capa es de vital importancia ya que es la que entra en contacto con el usuario final de la aplicación. Con el objetivo de que sea del agrado del usuario, tendremos que ser especialmente cuidadosos con el diseño y el estilo.

Todas las pantallas mostradas en una aplicación Android están formadas por Views. Un View es una clase que representa un elemento dentro de la interfaz de usuario (botones, cuadros de textos, listas, etc). Las distintas Views que componen una pantalla dentro de una aplicación Android se agrupan en objetos contenedores llamados ViewGroup, que es una clase abstracta que sirve de contenedor de Views. Existen diferentes tipos de ViewGroup (llamados Layout) en función de cómo se distribuyen los elementos que lo componen (de forma lineal, relativa o tabular).

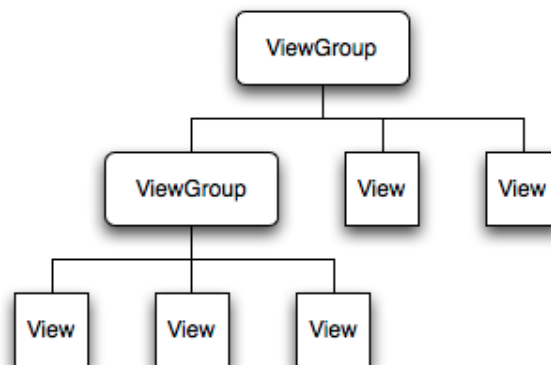


Figura 6.32: Diseño – View y Viewgroup

Una lista con los diferentes tipos de Views presentes en Android sería la siguiente:

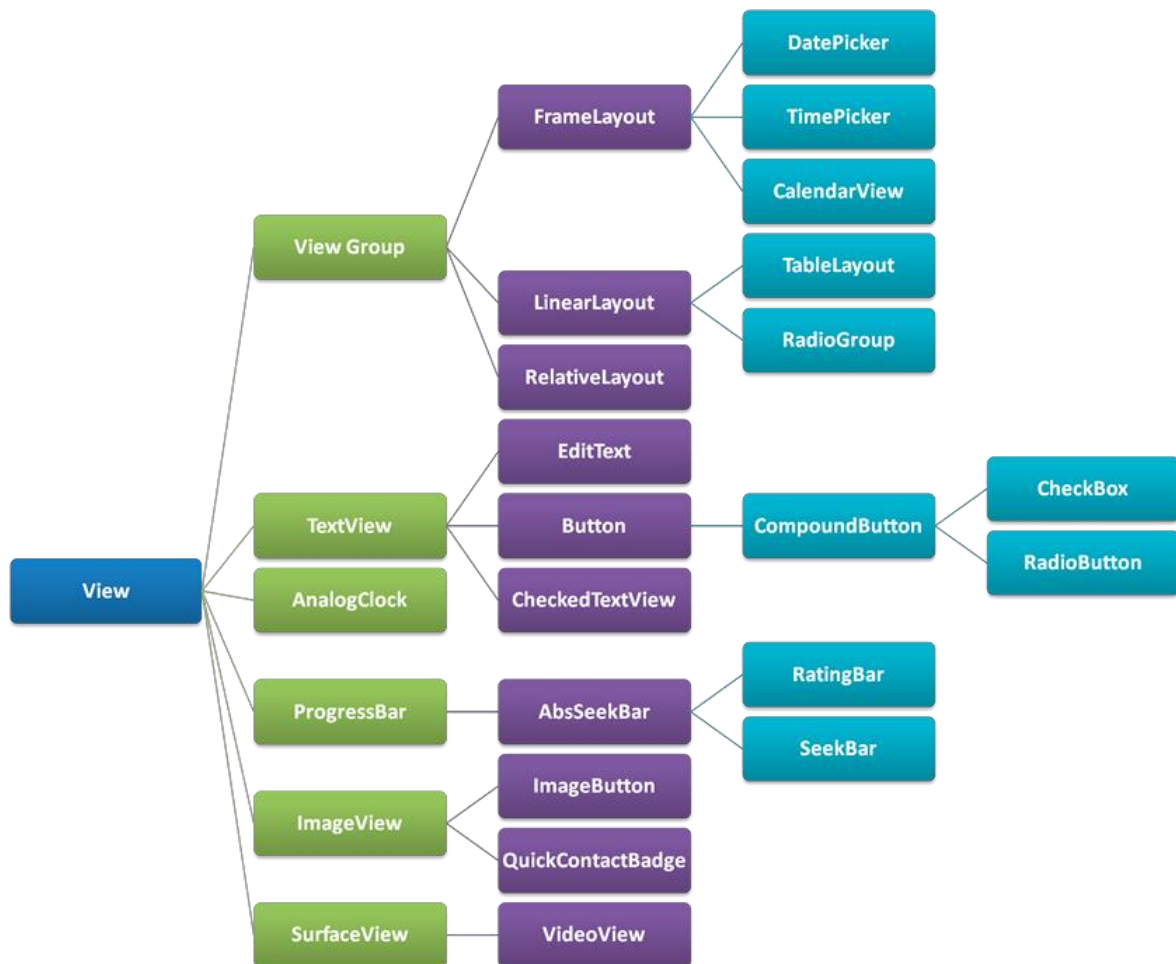


Figura 6.33: Diseño – Tipos de Views

Todos estos objetos se definen a través de un fichero xml que debe estar en la carpeta “/drawable/layout” dentro de la estructura del proyecto. De esta forma, para cada pantalla que tengamos en la aplicación deberemos tener un xml que defina su interfaz. Estos ficheros xml están compuestos por objetos View o ViewGroup de tal forma que los atributos del xml sean las diferentes propiedades de configuración del objeto (tamaño, color, texto...).

A continuación veremos el diseño de las principales pantallas de la aplicación y se describirán las características más importantes de cada una de ellas. El resto de pantallas posee un diseño similar a las aquí recogidas.

- **Pantalla Login**

- La palabra 'Moodle' se encuentra en un TextView con el valor del texto ya establecido.
- Los 3 campos para que el usuario introduzca sus credenciales son del tipo EditText. En el caso del 'Password', se indica que es una entrada de ese tipo para que oculte con '\*' todo lo escrito.
- Tanto 'Cuentas' como 'Login', son botones (Button).
- La cabecera que contiene 'Cuentas' está incluida en un RelativeLayout, mientras que el resto de elementos se incluyen en un LinearLayout.

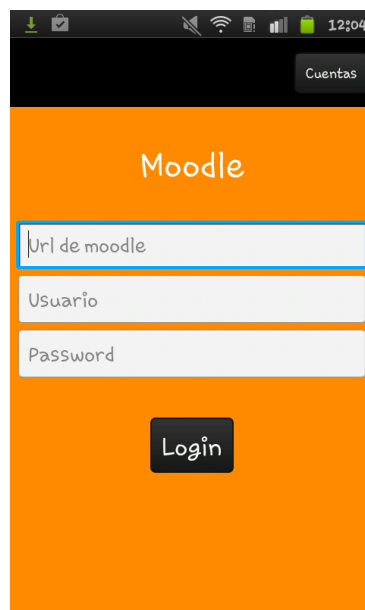


Figura 6.34: Diseño de pantalla – Login

- **Pantalla Login desde cuenta almacenada**

- 'Volver' es un botón incluido dentro de un LinearLayout
- Además, existe otro LinearLayout cuyo contenido es un ListView, que estará compuesto por todas las cuentas previamente almacenadas.
- Cada elemento de la lista se encuentra en un LinearLayout compuesta por un ImageView(imagen del usuario), TextView(nombres y apellidos) y un ImageView con el botón (flecha) para hacer login.

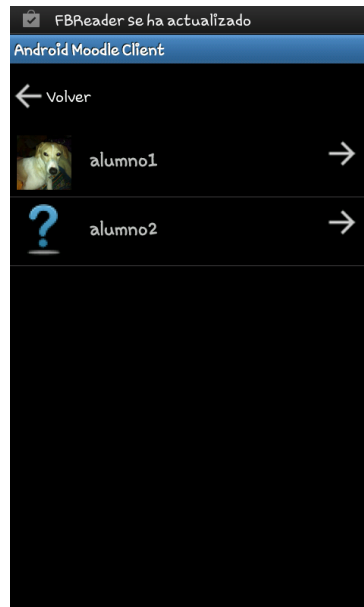


Figura 6.35: Diseño de pantalla – Login desde cuenta almacenada

- **Pantalla Principal**

- La parte superior de la pantalla está compuesta por un LinearLayout que contiene un ImageView con el icono de la aplicación, y un TextView en el que irá escrito el nombre del sitio al que el usuario se encuentra conectado.
- En la parte inferior se incluye un elemento de tipo TableLayout. Estará formado a su vez por 3 TableRows, 3 filas compuesta por 2 botones cada una, correspondientes a las diferentes secciones de la aplicación.



Figura 6.36: Diseño de pantalla – Menú Principal

- **Pantalla Subidas**

- Como se ha comentado en pantallas anteriores, el botón de ‘Volver’ se encuentra en un LinearLayout.
- El resto de la pantalla está englobado en otro LinearLayout.
- En la parte superior se observa otro LinearLayout que incluye la imagen (ImageView) y el texto (TextView) de ‘Subir’.
- El resto está formado por una TableLayout con 3 TableRows, cada una con un botón representando las acciones posibles.

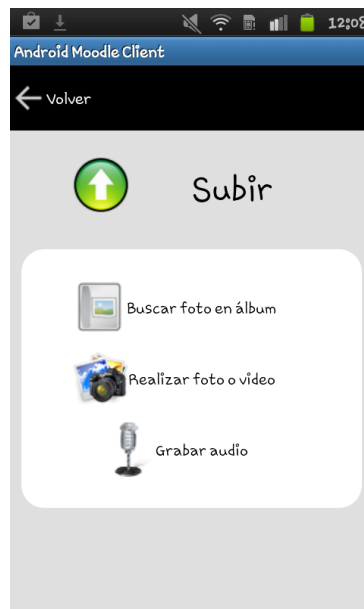


Figura 6.37: Diseño de pantalla – Pantalla Subidas

- **Pantalla Participantes**

- Volvemos a encontrarnos con el botón ‘Volver’ incluido en un LinearLayout.
- En la parte inferior existe otro LinearLayout formado por un ListView para listar cada uno de los participantes del curso.
- Cada elemento de la lista se encuentra en un LinearLayout con un TextView para indicar el nombre y apellidos del participante y un ImageView que contiene el icono (flecha) para acceder al detalle del participante.



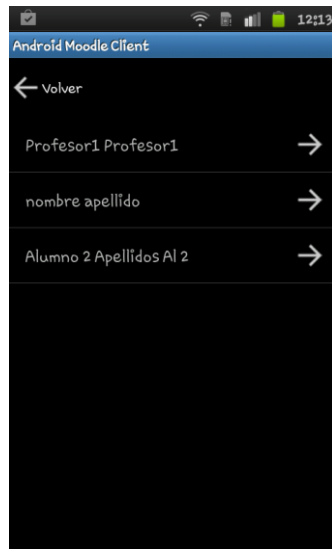


Figura 6.38: Diseño de pantalla – Listado de participantes

- **Pantalla Detalle Participante**

- Volvemos a encontrarnos con el botón ‘Volver’ incluido en un LinearLayout.
- La primera información relativa al participante viene recogida dentro de un LinearLayout. Contiene un ImageView con su imagen y un TextView con su nombre y apellidos.
- El resto de la información del usuario aparece en un TableLayout en el que se combinan TextView para las etiquetas y para reflejar la información.
- Por último aparecen 2 LinearLayout conteniendo los botones con las posibles acciones.



Figura 6.39: Diseño de pantalla – Detalle participante

- **Pantalla Cursos**

- Volvemos a encontrarnos con el botón ‘Volver’ incluido en un LinearLayout.
- En la parte inferior existe otro LinearLayout formado por un ListView para listar cada uno de los cursos.
- Cada elemento de la lista está formado por un TextView con el nombre del curso y un ImageView que contiene el icono (flecha) para acceder al detalle.

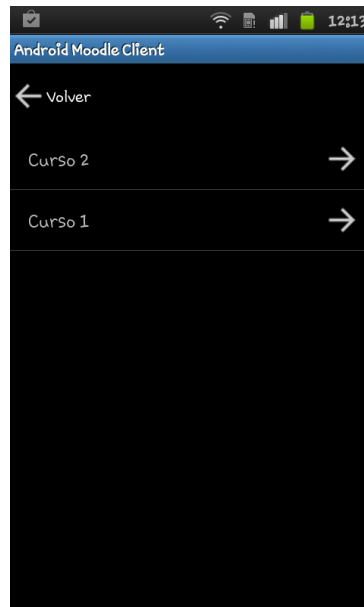


Figura 6.40: Diseño de pantalla – Lista de cursos

- **Pantalla Detalle Curso**

- Volvemos a encontrarnos con el botón ‘Volver’ incluido en un LinearLayout.
- Lo siguiente que aparece es un LinearLayout formado por un ListView. Los elementos de esa lista son las secciones en las que se divide el curso.
- Por cada sección, existe un LinearLayout con un TextView con el nombre la misma y otro TextView con el subtítulo. Además, incluyen un ListView con los distintos módulos que la componen.
- A su vez, un módulo está formado por un ImageView con su imagen, un TextView con su nombre y un ImageView con la acción a realizar.



Figura 6.41: Diseño de pantalla – Detalle curso

- **Pantalla Elementos Carpeta**

- Volvemos a encontrarnos con el botón ‘Volver’ incluido en un LinearLayout.
- Lo siguiente que aparece es un LinearLayout que contiene un TextView con el título que vemos y un ListView. Los elementos de esa lista son los que conforman la carpeta.
- Cada elemento está incluido en un LinearLayout con un TextView con su nombre y un ImageView con la acción a realizar.

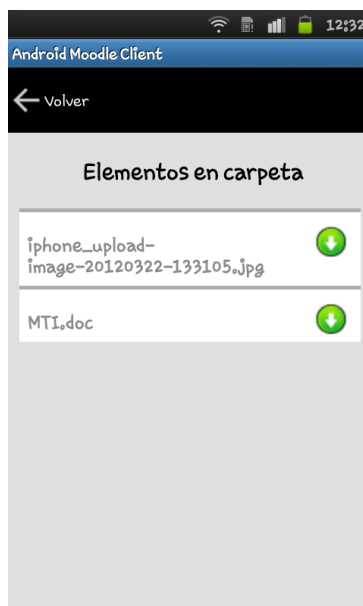


Figura 6.42: Diseño de pantalla – Elementos de una carpeta

- **Pantalla Estadísticas**

- Pantalla compuesta por un RelativeLayout.
- Incluye un TextView con el título y un LinearLayout con un tipo GraphicalView para dibujar el gráfico con los datos estadísticos. Para poder utilizar el tipo GraphicalView se utiliza la librería externa 'achartengine.jar'.

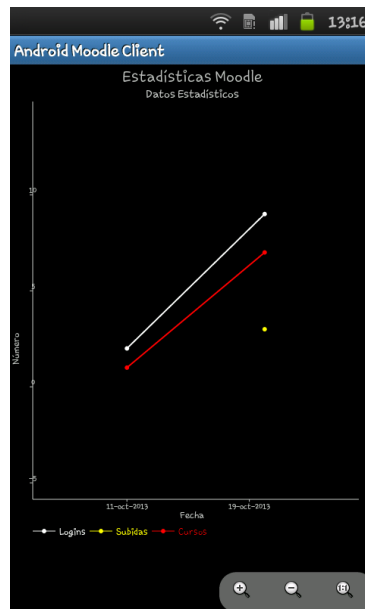


Figura 6.43: Diseño de pantalla – Estadísticas

- **Pantalla Web Moodle**

- Volvemos a encontrarnos con el botón 'Volver' incluido en un LinearLayout.
- Lo siguiente que aparece es un tipo WebView donde cargaremos las páginas relativas a la versión web del sitio Moodle.
- Por último, aparece un LinearLayout que contiene 4 RelativeLayout. Cada uno de ellos está formado por un botón que usaremos para la navegación (atrás, adelante, recargar y navegador).

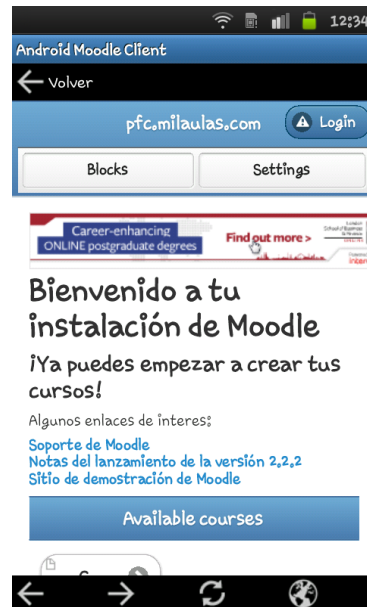


Figura 6.44: Diseño de pantalla – Versión Web

- **Pantalla Ayuda**

- Volvemos a encontrarnos con el botón ‘Volver’ incluido en un LinearLayout.
- Lo siguiente que aparece es un tipo WebView donde cargaremos las páginas relativas a la ayuda oficial de Moodle.
- Por último, aparece un LinearLayout que contiene 4 RelativeLayout. Cada uno de ellos está formado por un botón que usaremos para la navegación (atrás, adelante, recargar y navegador).

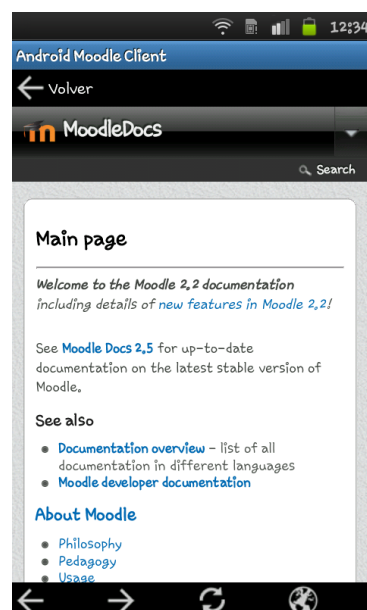


Figura 6.45: Diseño de pantalla – Ayuda

---

## ***Implementación***

---

### ***7.1 Instalación del entorno de trabajo***

En este manual se va a tratar la instalación paso a paso de las herramientas necesarias para comenzar a desarrollar nuestra aplicación sobre un sistema Android. Los principales componentes son el SDK de Android, el plugin ADT para eclipse, así como el propio IDE Eclipse.

En este caso se trabaja con el IDE Eclipse Galileo for Java Developers, por lo éste será el caso que se detalla para la instalación del SDK Android.

#### **7.1.1 Requisitos del sistema**

A partir de este momento vamos a tratar todo lo necesario para preparar nuestro entorno de trabajo y poder empezar con la fase de desarrollo. Para ello, el primer paso es revisar los requisitos. Dichos requisitos, según la propia documentación de Android, son los siguientes:

- Sistema Operativo:
  - Windows XP (32-bit), Vista (32 o 64-bit), o Windows 7 (32 o 64-bit)
  - Mac OS X 10.5.8 o superior (solo x86)
  - Linux (probado en Linux Ubuntu Hardy Heron)
  
- Entorno de desarrollo: (Eclipse IDE)
  - Eclipse 3.4 (Ganymede) o 3.5 (Galileo)
  - Eclipse JDT plugin
  - JDK 5 or JDK 6
  - Android Development Tools plugin (opcional)

- No compatible con Gnu Compiler para Java (gcj)
- Otros entornos de desarrollo:
  - Apache Ant

### 7.1.2 Instalación de Eclipse

Aunque la instalación del IDE no forma parte del proyecto, vamos a proceder a su instalación, que es fácilmente abordable desde su propia web (<http://www.eclipse.org>). En la sección "Download", se elige la versión deseada, que en nuestro caso será Eclipse IDE for Java Developers para Windows 32 bits, con un tamaño de 99MB.

Una vez descargado, basta descomprimir y ejecutar "eclipse.exe" para comenzar a usar la aplicación.

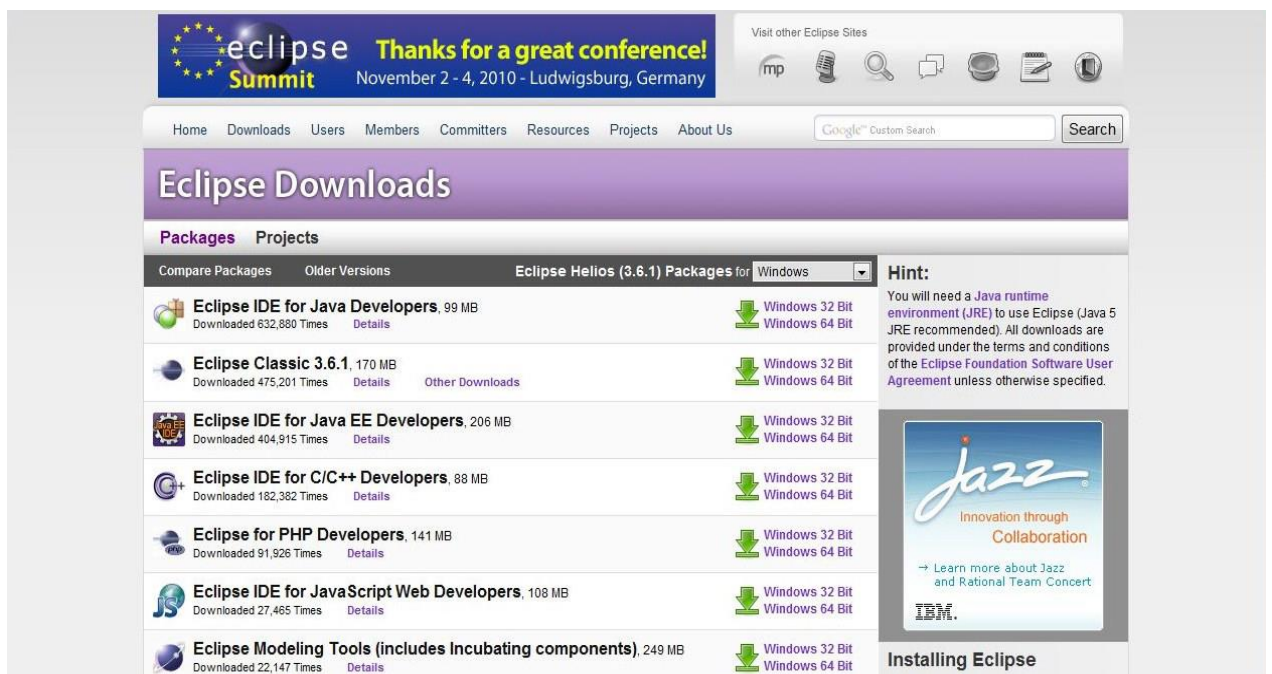


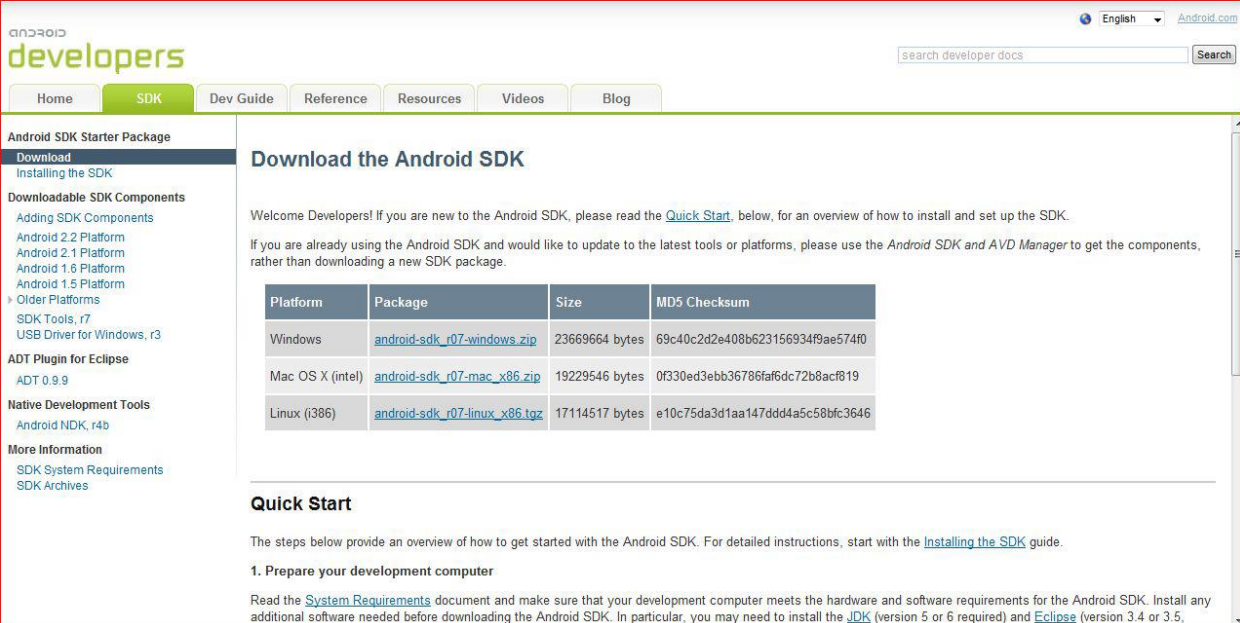
Figura 7.1: Descarga de Eclipse

### 7.1.3 Instalación del SDK Android

Entre los números recursos disponibles en la web de Android (<http://developer.Android.com>) tenemos todas las versiones del SDK, en la sección "SDK". En nuestro caso elegimos la

versión para Windows, con un tamaño de 23MB. Una vez completada la descarga, descomprimos en la ubicación deseada.

Antes de proseguir, es recomendable almacenar la ruta de la subcarpeta "tools" en la variable de entorno "path" de nuestro sistema. Para ello hacemos clic con el botón derecho en "Equipo" para acceder a las Propiedades del sistema. En configuración avanzada podemos acceder a dichas variables. Basta buscar la variable "path" en la lista y hacer doble clic para editar. Al valor actual debemos añadir nuestra ruta, precedida por ";", para conservar los valores anteriormente almacenados.



The screenshot shows the 'Download the Android SDK' page on the Android Developers website. The sidebar on the left contains links for 'Android SDK Starter Package', 'Download', 'Installing the SDK', 'Downloadable SDK Components', 'Adding SDK Components', 'Android 2.2 Platform', 'Android 2.1 Platform', 'Android 1.6 Platform', 'Android 1.5 Platform', 'Older Platforms', 'SDK Tools, r7', 'USB Driver for Windows, r3', 'ADT Plugin for Eclipse', 'ADT 0.9.9', 'Native Development Tools', 'Android NDK, r4b', 'More Information', 'SDK System Requirements', and 'SDK Archives'. The main content area has a heading 'Download the Android SDK' and a welcome message. Below the message is a table with the following data:

Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-sdk_r07-windows.zip</a>	23669664 bytes	69c40c2d2e408b623156934f9ae574f0
Mac OS X (intel)	<a href="#">android-sdk_r07-mac_x86.zip</a>	19229546 bytes	0f330ed3ebb36786faf6dc72b8acf819
Linux (i386)	<a href="#">android-sdk_r07-linux_x86.tgz</a>	17114517 bytes	e10c75da3d1aa147ddd4a5c58bfc3646

Below the table is a 'Quick Start' section with the text: 'The steps below provide an overview of how to get started with the Android SDK. For detailed instructions, start with the [Installing the SDK](#) guide.' and a numbered list starting with '1. Prepare your development computer'. At the bottom, it says: 'Read the [System Requirements](#) document and make sure that your development computer meets the hardware and software requirements for the Android SDK. Install any additional software needed before downloading the Android SDK. In particular, you may need to install the [JDK](#) (version 5 or 6 required) and [Eclipse](#) (version 3.4 or 3.5).'

Figura 7.2: Descarga de SDK Android

Esto nos facilitará el acceso mediante símbolo del sistema a algunas configuraciones del SDK Android por comandos, no siendo necesario escribir la ruta completa de la ubicación de las utilidades.

### 7.1.4 Instalación del plugin ADT de Eclipse

Usando Eclipse, este plugin nos facilitará tareas de programación, simulación y depuración de nuestras aplicaciones. Para descargarlo, basta con acceder a "Install new software" del menú "Help" e introducir la dirección donde se ubica (<https://dl-ssl.google.com/Android/eclipse>) como directorio de trabajo. Aunque por razones de seguridad es preferible usar el protocolo https, en caso de problemas también podría usarse http.



A continuación debe ofrecernos la instalación de "Developer Tools", y desplegando el árbol podremos ver que se compone de "Android DDMS" y "Android Developer Tools". Elegimos ambas herramientas y tras aceptar la correspondiente licencia en el siguiente paso, comienza la descarga del nuevo software.

Una vez descargado e instalado hay que configurar las nuevas utilidades para que Eclipse reconozca el SDK Android. Para ello, en el menú "Window/Preferences" elegimos la opción Android. En el cuadro de dialogo mostrado debemos introducir la ruta donde fue instalado el SDK en el paso anterior, por ejemplo C:\Android-sdk-windows. Si todo el proceso se ha realizado correctamente, tras pulsar el boton "Apply" nos aparecerán todas las versiones de las APIs incluidas en nuestro SDK. Aceptamos los cambios y ya podremos empezar a desarrollar para Android.

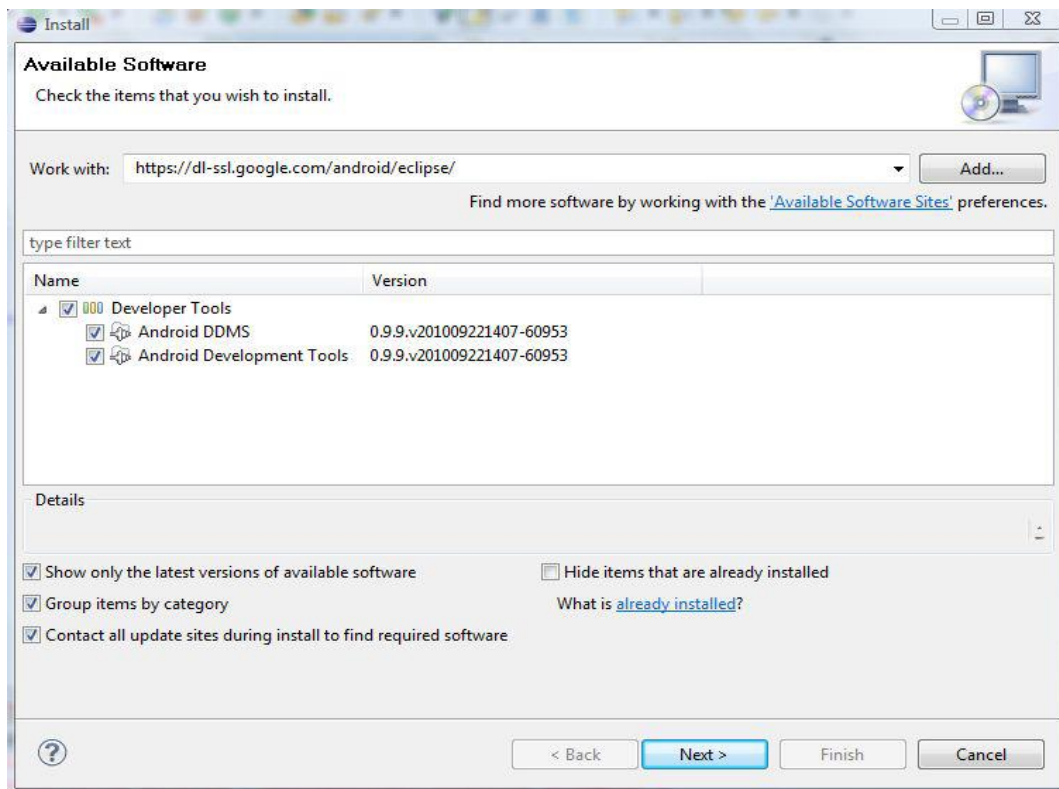


Figura 7.3: Instalación de Android Developer Tools

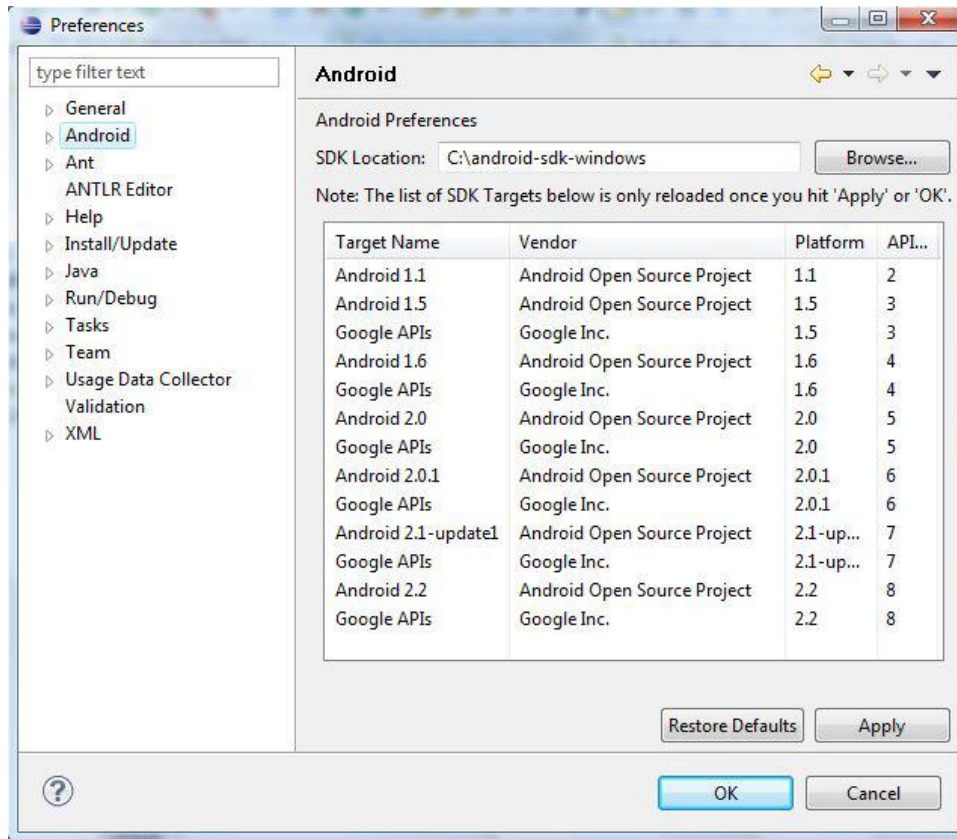


Figura 7.4: Especificar ubicación del SDK

Sin embargo, para completar la puesta en marcha de las utilidades que nos ofrece el plugin, todavía nos queda por crear nuestro simulador, o simuladores, para probar nuestras aplicaciones. Esto evitará tener que acceder a nuestro terminal para testear cada cambio realizado en el código fuente desarrollado.

Para ello accedemos al menú "Window\Android SDK and ADV Manager", que nos mostrará una ventana del mismo nombre. Ahí, seleccionando la opción "Virtual Devices" podremos crear fácilmente nuestros dispositivos haciendo clic en "new...". Basta elegir el nombre deseado, elegir la versión de la API deseada y añadir la compatibilidad hardware que vayamos a necesitar. Tras hacer clic en "Create ADV" nuestro dispositivo aparecerá en el listado.

**Opcional.** El plugin de desarrollo crea de forma automática una carpeta llamada ".Android" dentro de la carpeta "Mis documentos", donde se almacenan, entre otros cheros de configuración, nuestros dispositivos virtuales creados. Si la configuración de esta carpeta es la definida por defecto en nuestro sistema operativo Windows, no tendremos ningún problema, pero si hemos cambiado la ruta de acceso habrá que indicar a Eclipse y al plugin de Android nuestra ubicación personalizada. La forma más sencilla es creando una nueva variable de entorno, procediendo de la misma forma que cuando añadimos a la variable "Path" la ruta de nuestro SDK Android. En este caso la diferencia sería que crearemos una nueva variable en

lugar de modificar una ya existente. La llamaremos "ANDROID SDK HOME", y su valor será la ruta de nuestra carpeta "Mis documentos", por ejemplo "D:\documentos".

Por último, señalar que en caso de que aparecieran nuevas versiones del plugin podremos acceder a ellas mediante la herramienta "Check for updates" del menú "Help".

Si lo que queremos es actualizar el SDK basta con descargar el nuevo paquete y cambiar, si fuera necesario, la ruta de instalación especificada en el plugin de Android.

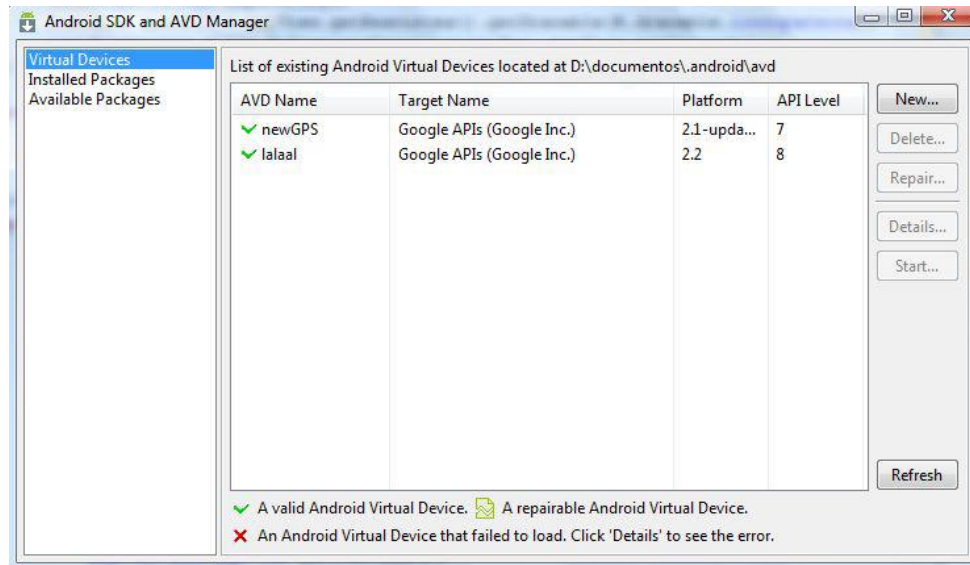


Figura 7.5: ADV Manager

## 7.2 *AndroidManifest.xml*

Este fichero se genera automáticamente al crear un proyecto de tipo 'Android project' en el entorno de desarrollo eclipse y su función principal es definir algunas de las especificaciones de la aplicación que estamos desarrollando (intents que la componen, icono de la aplicación, permisos, etc).

A continuación, se detallan los atributos que hemos tenido que definir para componer el AndroidManifest.xml de nuestra aplicación así como su significado:

Atributo	Valor	Significado
manifest /package	com.quique.pfc	Sirve para indicar al compilador en qué paquetes están las clases de nuestra aplicación. Además, el paquete será nuestro identificador único en Google Play
manifest/versionCode	1	Indica a los desarrolladores la versión del producto. Sirve también para manejar distintas versiones de la aplicación y llevar un mejor control sobre las mismas en Google Play.
manifest/versionName	1.0	Versión de la aplicación que se verá en la tienda de aplicaciones Google Play.
uses-sdk Android:minSdkVersion	7	Indica la mínima versión de la API (y por tanto, del sistema operativo) bajo la que puede funcionar la aplicación
uses-permission	Android.permission.INTERNET Android.permission.READ_CONTACTS Android.permission.WRITE_CONTACT Android.permission.RECORD_AUDIO Android.permission.EXTERNAL_STORAGE	Indica los permisos para los cuales se va a pedir permiso al usuario antes de que se realice la instalación de la aplicación.  En el caso del cliente Moodle son necesarios los permisos de:  - Salida a internet: para conectar con el servidor de

		<p>Moodle</p> <ul style="list-style-type: none"> <li>- Lectura/escritura de contactos: para la funcionalidad de añadir a participantes Moodle como contactos del teléfono</li> <li>- Grabar audio: para la funcionalidad de subir audio a tu espacio personal de Moodle.</li> <li>- Almacenamiento externo: para la bajada de los ficheros correspondientes al detalle de los cursos</li> </ul>
application/icon	@drawable/Moodle	Referencia al fichero (dentro de la carpeta drawable) que será el icono de la aplicación
application/label	@drawable/app_name	Nombre de la aplicación. app_name es una referencia a una entrada al fichero de cadenas (strings.xml)
activity Android:name=	Activities.AudioActivity Activities.AyudaActivity	<p>Definición de las diferentes pantallas que componen la aplicación. Para la pantalla de inicio (InicioActivity en nuestro caso), se debe añadir el atributo</p> <p>Android.intent.action.MAIN, de esta forma Android sabe qué actividad cargar al ejecutar la aplicación por primera vez.</p>

Tabla 7.1: Tabla de atributos AndroidManifest.xml

### 7.3 Internacionalización

Dado el hecho de que la aplicación desarrollada es un cliente genérico para cualquier sitio Moodle a partir de la versión 2 y que el sistema Moodle es el más extendido en cuanto a plataformas educativas online, la aplicación puede ser fácilmente adaptada y traducida a otros idiomas, sin tener que cambiar el código fuente de la misma.

Para ello, Android proporciona una estupenda infraestructura, ya que el sistema operativo y la API de programación que proporciona están pensados para que las aplicaciones sean internacionalizadas.

Para la internacionalización de una aplicación, en la carpeta de recurso que incorporan todas las aplicaciones (*res*), debe existir una subcarpeta *values* que contendrá un único archivo: *strings.xml*. Este fichero debe contener todas las cadenas de textos utilizadas dentro del código del sistema: cuadros de texto, valores de los botones, títulos, mensajes de confirmación, mensajes de espera, etc.

En la siguiente tabla se muestra un ejemplo de definición de una cadena dentro del propio fichero:

```
<string name="foto_subida_ok">Foto subida correctamente al servidor</string>
```

Tabla 7.2: Tabla de ejemplo de definición de una cadena

Las cadenas de textos pueden ser referenciadas desde 2 puntos dentro del código de una aplicación Android: los ficheros *xml* que definen las vistas y dentro del propio código de la aplicación.

- Para referenciar una cadena de texto dentro de un *xml* que define un *layout* utilizamos la etiqueta *@string* seguida del nombre de la cadena que queremos usar y que se encuentra definida en el fichero *string.xml*. De esta forma conseguimos también que si cambia una cadena de texto (por ejemplo, descubrimos un error ortográfico) no es necesario que se vuelva a escribir en todas las vistas donde se utiliza, que potencialmente pueden ser muchas, sino que bastaría con cambiar la definición de la cadena en el *strings.xml*. Ejemplo:

```
android:text="@string/login"
```

Tabla 7.3: Tabla de ejemplo de referencia a cadena desde *layout*

- Si queremos utilizar una cadena definida en el fichero 'xml' dentro del código java es necesario invocar a los métodos '**getResources().getString()**' que tiene el objeto '*Context*'. El objeto '*Context*' es el contexto de la aplicación y todas las '*Activities*' implementadas heredan de él. Ejemplo:

```
//enviar_servidor es el nombre de la cadena en el fichero strings.xml
getResources().getString(R.string.enviar_servidor);
```

Tabla 7.4: Tabla de ejemplo de uso de cadena desde el código

Una vez hemos definido nuestro fichero de cadenas, si queremos traducir la aplicación a otro idioma, sólo tenemos que copiar el fichero '*strings.xml*', traducir sus textos, y almacenarlos en una nueva carpeta '*values-XX*' donde XX es el código del idioma al que se ha traducido. En nuestro caso, se han incluido carpetas para traducir la aplicación al inglés y el francés (EN y FR respectivamente).

De esta forma, al arrancar la aplicación, Android detecta el código del idioma del teléfono e intenta hacer un reemplazo de las cadenas buscando el fichero '*string.xml*' en la carpeta '*values*' del código de idioma. En caso de no encontrarlo, el sistema recurrirá a la carpeta de valores por defecto.

## 7.4 Problemas encontrados

Junto al desconocimiento inicial sobre la programación para dispositivos móviles y más concretamente en la plataforma Android, los principales problemas encontrados a la hora del desarrollo han sido dos: el desconocimiento sobre los servicios web de Moodle y los problemas de rendimiento de la aplicación.

Respecto al primer problema, la documentación tanto de la configuración como de la descripción de los métodos ofrecidos por los 'Mobile Web Services' utilizados era prácticamente nula. La interfaz web Services no sigue un criterio uniforme a la hora de nombrar los parámetros e incluso tampoco mantiene constante la URL a la hora de invocar a los servicios web. Por ejemplo, para obtener un token de acceso es necesario invocar a una url, para invocar a las distintas funciones de consulta, otra url, para subir un fichero otra url distinta...

Afortunadamente, existe un servicio web que devuelve todas las funciones disponibles para los 'Mobile web Services'. De esta forma, se ha podido investigar invocando a cada una de las funciones devueltas por el servicio web y analizando la salida ofrecida por este. Tras analizar la salida 'json' de cada uno de los métodos, se ha construido un objeto java equivalente para la posterior conversión.

El otro problema no previsto inicialmente en el análisis de la aplicación ha sido el rendimiento. Las redes móviles son considerablemente más lentas que las conexiones de red de los PCs y, por tanto, se notó que al realizar un alto número de peticiones al servidor, la aplicación se ralentizaba en exceso, provocando ocasionalmente fallos inesperados por 'timeouts'.

Concretamente, la pantalla que tenía un menor rendimiento es la que muestra el detalle de un curso y eso era debido a que la aplicación tenía que hacer una llamada 'http' al servidor de Moodle para traerse los iconos de cada uno de los módulos que componen el curso. Para solucionar esto, se depuró e inspeccionó la respuesta obtenida por el servicio web de consulta de cursos para obtener de forma externa todos los posibles iconos y así incluirlos como recursos en la aplicación.

Además, teniendo en cuenta que ni el listado de cursos disponibles para un usuario ni la lista de participantes de un curso es algo que deba cambiar con mucha frecuencia, se ha implementado un sistema de cacheo que mantiene los datos 'vivos' en el teléfono durante un tiempo (5 minutos) definido en una de las constantes de la aplicación.

## 7.5 Caché

Debido a que en la implementación inicial se detectó que el rendimiento de la aplicación era bajo, se hizo un estudio para detectar la causa de la lentitud del sistema. Se pudo observar que se hacían demasiadas conexiones a los Web Services de Moodle. Esas conexiones se hacían especialmente lentas (llegando a tardar más de 10 segundos en alguna ocasión) cuando se realizaban desde una red 3G, llegando incluso a producir errores de TimeOut en alguna ocasión. Con redes wifi los tiempos mejoraban sensiblemente.

Finalmente se optó por implementar algunos mecanismos de cache para no tener que recurrir constantemente al servidor Moodle a la hora de traernos los diferentes datos mostrados en pantalla. Tanto los cursos como sus participantes son un conjunto de datos que no suele cambiar con mucha frecuencia, por lo que en la lógica de negocio encargada de esas consultas a Moodle se mantienen los datos 'vivos' durante cinco minutos. De este modo se comprobó



que la aplicación funciona de manera mucho más fluida y se reducen bastante los tiempos, haciendo que las peticiones no superen los 3 segundos.

La manera de implementarlo es manteniendo una serie de estructuras de datos estáticas en la clase *'MoodleService.java'* y variables con una marca de tiempo que indiquen la última vez que se consultaron los datos:

```
//participantes
private static Map<String, ParticipanteDto> participantes =
    new HashMap<String, ParticipanteDto>();

//cursos
private ArrayList<CursoDto> cursos = new ArrayList<CursoDto>();

//ultimos accesos a la variables (en timestamps)
private static long cursos_ts = 0;
private static long participantes_ts = 0;
```

Tabla 7.5: Tabla Cache – variables timestamps

De esta forma, la clase *'MoodleService.java'* mantiene un mapa donde la clave es el identificador único del curso y el valor es una lista con los participantes que la componen. Además, mantiene la lista de los cursos disponibles en el sistema.

En los métodos de consulta, el primer paso es comprobar si el 'timestamp' actual difiere una cantidad de tiempo (5 minutos) de la variable que indica el último acceso. En caso de que la diferencia sea menor, se devuelven los datos cacheados y si ha pasado más tiempo, se utiliza el *'MoodleDAO'* para acudir al servidor de Moodle para traer los datos más actualizados:

```
public ArrayList<CursoDto> obtenerCursos(String token, String userId)
{
    MoodleDao mdao = new MoodleDao();
    if(cursos == null || cursos.size() == 0)
    {
        cursos = mdao.obtenerCursos(token, userId);
    }
    if(System.currentTimeMillis() - cursos_ts > Constantes.TIEMPO_CACHE)
    {
        cursos = mdao.obtenerCursos(token, userId);
        cursos_ts = System.currentTimeMillis();
    }
    return cursos;
}
```

Tabla 7.6: Tabla Cache – comprobación timestamps

Adicionalmente, se ha implementado un sistema similar a las sesiones utilizadas en la programación web que mantiene los datos más frecuentados por el usuario en una variable de la aplicación (una clase con atributos estáticos en nuestro caso). Cada vez que se hace login en la aplicación se limpia este objeto 'Sesion' y tras producirse el acceso se establecen en el objeto el token de usuario para el uso de los servicios web, su identificador único en base de datos, el contexto de la aplicación, etc.

## 7.6 Algoritmo AES

Una de las preocupaciones de la sociedad actual es la seguridad de la información. Cada vez existen más servicios a través de internet y, por tanto, deben existir ciertas medidas de seguridad que nos ayuden a preservar nuestra privacidad.

La aplicación desarrollada trabajará con datos de usuarios de cualquier sitio Moodle. Parte de esos datos se almacenarán en la base de datos que se guarda en el terminal. En caso de pérdida o robo del terminal, cualquiera podría acceder a la información. Es por ello que se ha decidido codificar el password que se almacena. El método elegido para esa codificación es el algoritmo AES, incluido en una clase Encriptación.java.

En la siguiente tabla pueden observarse las principales funciones de la clase:

```

/**Funciones para encriptar y desencriptar la cadena que se les pasa*/
public String encriptar(String texto_a_encriptar) throws Exception
{
    Key key = new SecretKeySpec(keyValue, ALGORITMO);
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.ENCRYPT_MODE, key);
    byte[] encrypted =
cipher.doFinal(texto_a_encriptar.getBytes("UTF-8"));
    String texto_encriptado = Base64.encodeToString(encrypted,
Base64.DEFAULT);

    return texto_encriptado;
}

public String desencriptar(String texto_encriptado) throws Exception
{
    Key key = new SecretKeySpec(keyValue, ALGORITMO);
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.DECRYPT_MODE, key);
    byte[] decodificar_texto =
Base64.decode(texto_encriptado.getBytes("UTF-8"), Base64.DEFAULT);
    byte[] desencriptado = cipher.doFinal(decodificar_texto);

    return new String(desencriptado, "UTF-8");
}

```

Tabla 7.7: Tabla AES – encriptación contraseña

---

# Pruebas

---

Se exponen en este capítulo las pruebas realizadas a la aplicación para verificar que funciona correctamente y que cumple con los requisitos marcados antes de comenzar su desarrollo. También se utilizan estas pruebas para incorporar algunas mejoras según la experiencia del usuario. Además, se hablará en este capítulo de Genymotion, la herramienta utilizada para las pruebas en la fase de desarrollo.

### 8.1 Genymotion

Conforme fui desarrollando la aplicación, realicé pruebas de las distintas secciones codificadas. Al principio esas pruebas las realicé en el emulador por defecto de Android integrado en el ADT Bundle. La experiencia con dicho emulador fue muy mala ya que era excesivamente lento, provocando demasiadas pérdidas de tiempo. Este hecho me hizo buscar alternativas para realizar pruebas sobre el código desarrollado y así llegué hasta Genymotion.

Genymotion es una máquina virtual con sistema operativo Android que corre sobre Virtual Box. No se trata de un emulador, sino que es una versión x86 del sistema operativo. Los resultados obtenidos fueron bastante satisfactorios desde que comencé a utilizarlo. Funciona mucho más rápido que el emulador. Por tanto, fue el método elegido para testear el código que iba desarrollando.

Puede descargarse Genymotion de su página oficial:

<http://www.genymotion.com/>

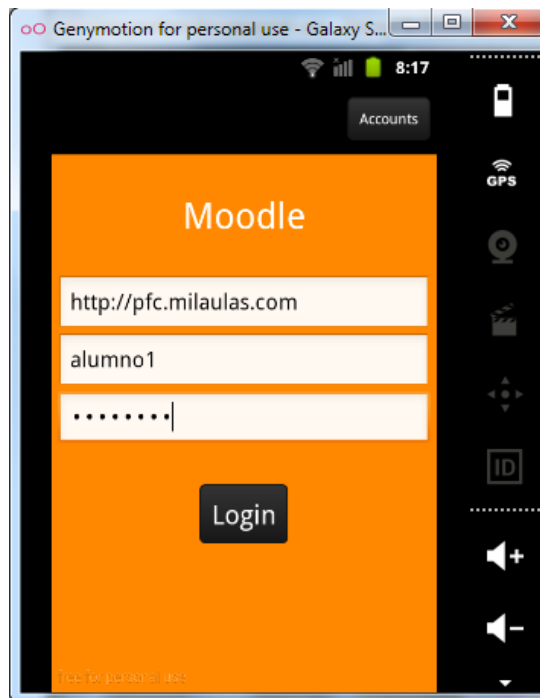


Figura 8.1: Genymotion

## 8.2 Junit

En Wikipedia, se define Junit como *“un conjunto de bibliotecas que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java”*.

Se trata de un framework (conjunto de clases) con el que podremos ejecutar clases Java de un modo controlado, ofreciendo la posibilidad de evaluar si los métodos que conforman la clase se comportan del modo esperado. Según un valor de entrada dado al método, se evalúa el valor de retorno esperado. En caso de que sea satisfactorio, Junit concluye que el test ha sido satisfactorio; en caso contrario, devolverá fallo en el método.

Las primeras comprobaciones realizadas se efectuaron sobre métodos independientes con un acoplamiento bajo con respecto a otras partes de la aplicación. Utilicé para ello la clase *“android.test.TestCase”*.

En el primer método, *testEncriptacion()* he comprobado que el algoritmo AES para la encriptación y desencriptación de cadenas funciona correctamente. En el siguiente método, *testAyudaMoodle()* he verificado que la ayuda de Moodle (una de las opciones de nuestra aplicación) se encuentra disponible para su consulta.

```

public void testEncriptacion() throws Exception{

    Encriptacion e = new Encriptacion();
    String contraseña_encriptada = e.encriptar("HolaMundo");
    String contraseña_desencriptada =
e.desencriptar(contraseña_encriptada);

    assertEquals("HolaMundo", contraseña_desencriptada);
}

public void testAyudaMoodle() throws Exception{

    HttpGet httpget = new
HttpGet("http://docs.moodle.org/22/en/Main_page");
    HttpClient httpclient = new DefaultHttpClient();
    HttpResponse httpresponse = null;

    try
    {
        httpresponse = httpclient.execute(httpget);
    }
    catch (Exception e)
    {
        fail("Error: Imposible acceder a la url especificada");
    }
}

```

Tabla 8.1: Tabla Junit – primer test

En la segunda batería de pruebas se probaron funciones que interactúan con otras partes de la aplicación, con un acoplamiento mayor. He utilizado la clase `android.test.AndroidTestCase` que extiende de la clase utilizada en las pruebas anteriores.

Durante estas pruebas se ha testado que es posible acceder a la base de datos en modo lectura/escritura y en modo lectura (*testBaseDatos()*). Por último, se ha comprobado mediante *testAgregarContactos()* que la aplicación tiene privilegios para escribir contactos en la agenda del terminal. De este mismo se podría comprobar el acceso a la cámara, acceso al almacenamiento externo...

```

public void testAgregarContactos()
{
    Context ctx = this.getContext();
    PackageManager pm = ctx.getPackageManager();
    int res =
pm.checkPermission("android.permission.WRITE_CONTACTS", "com.quique.pfc");
    assertEquals(res, PackageManager.PERMISSION_GRANTED);
}

```

```

public void testBaseDatos()
{
    try
    {
        nombreBD.openToRead();
        nombreBD.close();

        nombreBD.openToWrite();
        nombreBD.close();
    }
    catch (SQLException e)
    {
        fail("Error: Imposible acceder a la base de datos");
    }
}

```

Tabla 8.2: Tabla Junit – segundo test

### 8.3 UI Automator Viewer y uiautomator

Además de las pruebas unitarias sobre algunos de los componentes individuales que conforman la aplicación, también es importante realizar pruebas sobre el comportamiento de la interfaz de usuario de la aplicación cuando se ejecuta sobre un dispositivo. Con estas pruebas podemos asegurar que la aplicación devuelve la salida de interfaz correcta en respuesta a una secuencia de acciones ejecutadas por el usuario. En mi caso ejecutaré pruebas de forma manual y verificaré que la aplicación se comporta como se espera.

Ambas herramientas vienen incluidas en la SDK de Android y nos servirán para apoyar las pruebas de interfaz de usuario.

Con el fin de escanear y analizar gráficamente los componentes de la interfaz gráfica de la aplicación, he utilizado UI Automator Viewer. Dicha herramienta nos indicará cómo está organizada jerárquicamente la pantalla, además de mostrarnos los atributos de cada uno de sus componentes. Toda la información proporcionada por esta herramienta será de mucha utilidad posteriormente para las pruebas a ejecutar con uiautomator.

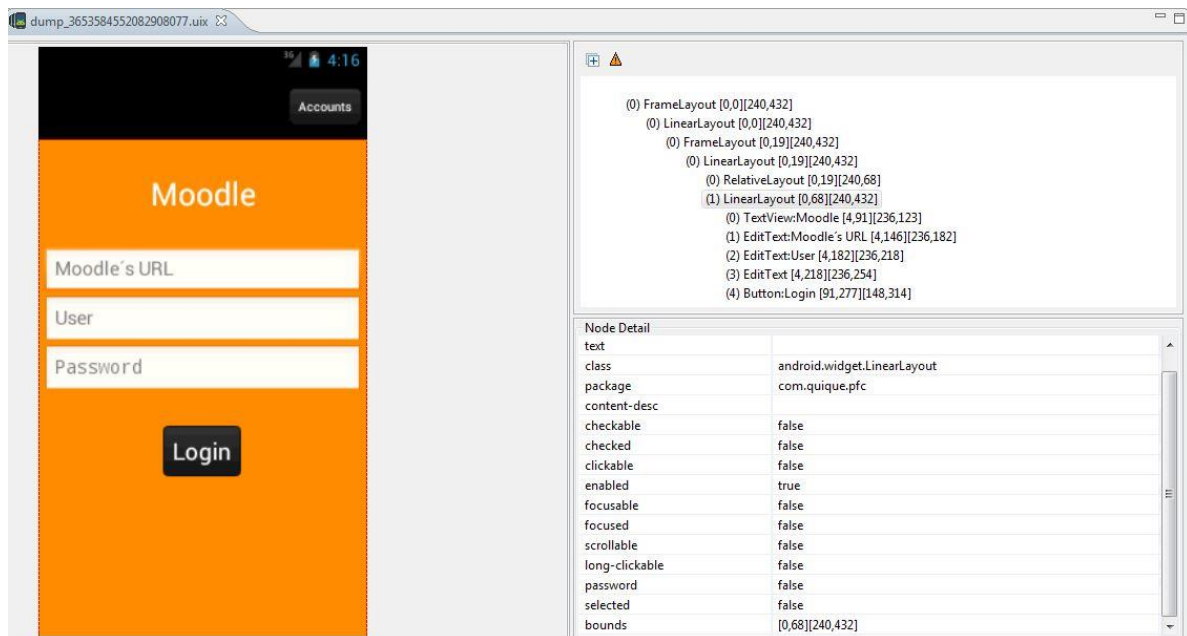


Figura 8.2: UI Automator Viewer

Gracias a los datos obtenidos en la herramienta ejecutada anteriormente, escribiré pruebas automatizadas sobre la interfaz de usuario con uiautomator. En la siguiente tabla se muestra el código en el que se prueba el caso de login rellenando todos los campos que aparecen en la pantalla inicial de la aplicación y se llega al menú principal.

```
public void testDemo() throws UiObjectNotFoundException {

    //Se pulsa el botón HOME del Terminal
    getUiDevice().pressHome();

    //Iniciamos la aplicación
    UiObject moodle_app = new UiObject(new UiSelector().text("Android
Moodle Client"));
    moodle_app.clickAndWaitForNewWindow();

    //Establecemos el valor de la url
    UiObject url = new UiObject(new
UiSelector().resourceId("com.quique.pfc:id/editText1"));
    url.setText("http://pfc.milaulas.com");

    //Establecemos el valor del usuario
    UiObject usuario = new UiObject(new
UiSelector().resourceId("com.quique.pfc:id/editText2"));
    usuario.setText("alumno1");

    //Establecemos el valor del password
    UiObject pass = new UiObject(new
UiSelector().resourceId("com.quique.pfc:id/editText3"));
    pass.setText("Alumno1");
}
```

```

    UiObject login_btn = new UiObject(new
    UiSelector().resourceId("com.quique.pfc:id/button_login"));

    //Hacemos click en el boton de login y esperamos a la nueva ventana
    login_btn.clickAndWaitForNewWindow();

    //Confirmamos que estamos en la pantalla principal de la aplicación
    UiObject login_ok = new UiObject(new
    UiSelector().resourceId("com.quique.pfc:id/boton_estadistica"));

    assertTrue(login_ok.exists());

}

```

Tabla 8.3: Tabla uiautomator

Una vez escrito el código, los pasos para la ejecución de los test serían:

- Generar el fichero de configuración del proyecto en curso ('build.xml')
- Compilar el proyecto de prueba para generar el '.jar' que ejecutaremos en el terminal
- Metemos el fichero generado en el terminal
- Ejecución de la prueba

El resultado será ver como en el terminal se muestran las pantallas implicadas en la prueba y cómo se rellenan de manera automática los datos solicitados al usuario. Finalmente, se muestra en el terminal si la prueba se ha ejecutado correctamente o no.

## 8.4 Pruebas y validaciones

### 8.4.1 Plan de pruebas

Con el fin de verificar que nuestra aplicación realiza una correcta gestión de los datos, se realizó la siguiente batería de pruebas:

- **Pruebas de caja negra independientes:** se realizan sobre clases independientes y que interactúan directamente sobre la base de datos. Lo importante en este tipo de pruebas no es la implementación, sino los datos que se proporcionan a la entrada y los datos que se obtienen a la salida.



- **Pruebas de caja negra sobre subsistemas:** se realizan sobre clases asociadas de alguna manera. Se verifica el paso de información de una clase a otra y la cohesión entre ambas.
- **Pruebas de caja blanca:** consistentes en recorrer todas las posibles opciones que engloba un módulo o clase para asegurarnos de que todas tienen el funcionamiento esperado.
- **Pruebas del sistema:** se prueba la buena comunicación entre los conjuntos de subsistemas o conjunto de clases agrupadas.

#### 8.4.2 Especificación del diseño de las pruebas

Las pruebas realizadas a la aplicación se han dividido en 2 fases:

- **Durante el desarrollo de la aplicación:** esta fase alberga todo tipo de pruebas relacionadas con las clases independientes. Para realizarlas, los datos de entrada utilizados eran dependientes del módulo que se estaba codificando en esos momentos. El conjunto de pruebas hacía referencia tanto al tratamiento de la información por parte de la aplicación como a la lectura y escritura en la base de datos. Con ello conseguimos evitar ir arrastrando errores desde el comienzo. En esta fase se utilizó Junit.
- **Finalizada la aplicación:** estas pruebas abarcan desde la instalación de la aplicación en diversos terminales hasta el chequeo de todas las funcionalidades que es capaz de realizar.

Una vez habíamos probado la instalación en varios terminales, se probaron los distintos métodos para logarnos a un sitio Moodle. Inicialmente se introdujeron los datos requeridos para acceder a varios sitios de pruebas, y posteriormente se accedió a los sitios desde las cuentas ya almacenadas.

El siguiente paso fue probar el módulo de subida de ficheros del usuario a su espacio privado. La primera opción fue la de seleccionar una foto almacenada en la galería de imágenes de nuestros terminal y subirla. Seguidamente se probó la realización de fotos y vídeos y subida de los mismos. Por último se verifica el funcionamiento de la grabación y subida de sonidos.

A continuación se accede al módulo de participantes. Se listan los distintos cursos a los que pertenece el usuario y se verifica la lista de participantes en un par de ellos. Se accede a la información detallada de uno de los participantes, se verifica que los datos

mostrados son correctos y que las posibles acciones a realizar tienen el funcionamiento esperado. Entre ellas se encuentran la de agregar los datos de participante a la agenda del terminal y la de escribirle un mensaje privado.

Para el testeo de los cursos asignados a un usuario, vemos primero que se listan todos los correspondientes y luego accedemos al detalle de cada uno. En el detalle, se verifica que aparezcan todas las secciones y módulos que las componen. Se prueba la descarga de ficheros al terminal y la apertura de aquellos que son posible visualizar porque el terminal tiene el visor correcto (pdf, Word, imágenes, urls...). Por último, se verifica también que se muestran los elementos de un tipo carpeta.

Las siguientes secciones que se prueban son la de la versión web de la aplicación y la de la ayuda oficial de Moodle, obteniendo resultados satisfactorios.

Finalmente, y debido a que ya se han registrado bastante actividad, se procede a probar la sección de datos estadísticos. Se dibujan correctamente los datos pertenecientes al número de login, de acceso a cursos y de ficheros subidos según la fecha en la que se producen.

### 8.4.3 Especificación de los casos de pruebas

Con el objetivo de realizar pruebas más específicas, se facilitó la aplicación a 2 grupos de usuarios bien diferenciados:

- *Personas con perfil bajo-medio en informática y conocimientos nulos en programación:* en este grupo de personas está formado por usuarios que conocen perfectamente qué debe hacer la aplicación y explotan todas sus funcionalidades, pero que desconocen por completo cómo lo hace. Desde el primer momento encontraron con facilidad las distintas funciones ofrecidas, la aplicación les resultó bastante útil. Como posible mejora, indicaron que sería mejor incluir un botón ‘Volver’ en todas las pantallas, ya que no les resultaba cómodo utilizar en todo momento el del terminal.
- *Personas con perfil alto en informática y amplios conocimientos en programación:* a este grupo se le facilita el código de la aplicación para que hagan una revisión más exhaustiva. Además de sugerir varias mejoras en el código para optimizar algunas funciones, indican 2 cambios importantes: dotar a la aplicación de un aspecto visual más profesional y cambiar el orden de las opciones del menú principal. Se aplican todas las sugerencias.

#### 8.4.4 Especificación de los procedimientos de pruebas

Además de los testeos realizados sobre la aplicación utilizando Genymotion, se han realizado pruebas sobre diferentes tipos de terminales, entre los que destacamos tabletas y móviles. El modelo de tableta probado fue el *Samsung Galaxy Tab*, mientras que de móvil se probaron diferentes modelos de *Samsung Galaxy (Ace, S2, y S3)* y de *HTC (One y Desire)*.

En cuanto a versiones de Android, se ha testeado la aplicación sobre *Jelly Bean v4.3*, *Ice Cream Sandwich v4.0* y *Gingerbread v2.3*.

Además, se realizaron pruebas sobre un terminal virtual con la versión mínima soportada por nuestra aplicación, *Android 2.2*.

Destacar que no se detectaron problemas en ninguno de los casos expuestos anteriormente.

#### 8.4.5 Documentación de la ejecución de las pruebas

- *Resultado de las pruebas*: la mayoría de errores detectados estaban ocasionados por pequeños fallos a la hora de llevar a cabo la implementación, siendo normales en esa fase. Los errores producidos por la aplicación en tiempo de ejecución fueron un poco más complicados de resolver. Lo más difícil de detectar fueron los errores producidos en el paso de información entre los distintos subsistemas que forman parte de la aplicación.
- *Informe de incidentes ocurridos*: no se han registrado incidentes durante toda la fase de pruebas. Sería aconsejable que a partir de ahora, todo el cambio sobre el código quedara registrado, incluida la persona que lo realiza. De este modo, podríamos llevar un control de versiones de la aplicación con los diferentes cambios introducidos.

---

# Conclusiones

---

Tras la detallada explicación llevada a cabo en capítulos anteriores, a continuación se hablará sobre las opciones que presenta el proyecto realizado y el posible enfoque que podrían tener futuros trabajos basados en la aplicación desarrollada. Además se expondrán las conclusiones propias a las que se ha llegado después de todo este proceso.

### 9.1 Aspectos generales

Tras concluir de manera definitiva el desarrollo del presente proyecto fin de carrera, es el momento de hacer balance del resultado final repasando la trayectoria recorrida para obtener conclusiones, tanto en el aspecto técnico y profesional como en el ámbito personal, además de las habilidades adquiridas durante este tiempo en diversas tecnologías y herramientas.

A lo largo del proyecto se ha conseguido obtener un amplio conocimiento de la plataforma Android, su funcionamiento interno, la manera que tiene de gestionar los recursos de un dispositivo móvil, además de aprender a desarrollar aplicaciones para dicha plataforma. Todo esto ha sido posible gracias a la extensa documentación que Google pone a disposición de los desarrolladores, aunque cabe destacar el importante trabajo de investigación que se ha realizado al respecto.

Por otro lado, tras haber cursado la titulación de Ingeniería Técnica en Informática de Gestión y haber tenido la oportunidad de hacer uso de muchos de los conocimientos adquiridos durante la carrera, el reto que suponía realizar el proyecto fin de carrera utilizando Java como lenguaje de programación, no ha hecho más que motivar el aprendizaje de dicho lenguaje, aun sabiendo la dificultad inherente de adquirir dicho conocimiento. Por lo tanto, tras los resultados obtenidos, y aunque la mayor dificultad encontrada ha sido instruirse en este lenguaje, creo oportuno decir que este objetivo se ha alcanzado con éxito.

En cuanto al resultado obtenido en la aplicación desarrollada, aunque existen ciertos puntos que podrían mejorarse sensiblemente, el balance final es muy positivo ya que se ha creado una

herramienta funcional que permite exprimir bastantes de las opciones que ofrece un sitio Moodle.

## 9.2 Conocimientos adquiridos

Para el desarrollo del proyecto se ha utilizado el entorno de desarrollo integrado (IDE) Eclipse [1] que nos permite programar en el lenguaje de programación Java [2]. Debido a que nuestra aplicación está destinada a dispositivos Android [3], se ha utilizado un kit de desarrollo de software (SDK) [4]. Además, con el objetivo de facilitar las tareas de programación, simulación y depuración, se ha utilizado el plugin ADT (Android Development Tools) [5]. En cuanto a librerías externas, se han utilizado google-gson [6] para realizar la conversión entre objetos json y Java y la librería AndroidPlot [7] para la creación de las gráficas estadísticas. Los datos de la aplicación son almacenados usando SQLite [8], sistema de gestión de base de datos ligero que administra el propio terminal. Las pruebas de ejecución de la aplicación se realizaron a través de Genymotion [18], mientras que para las pruebas de comportamiento de la interfaz de usuario se utilizó uiautomator [19]. Ninguna de estas herramientas fue vista durante la titulación.

Con el objetivo de conocer bien todas las opciones de las que constaría la aplicación, hubo que ampliar los básicos conocimientos sobre Moodle [9] que ya poseía.

Finalmente, para confeccionar la documentación se han aplicado muchos de los conocimientos adquiridos durante la titulación sobre ingeniería del software, especialmente orientada a objetos. Se han realizado tanto el análisis como el diseño de la aplicación al completo. Para ello, se ha utilizado la herramienta BOUML [10] con la que se ha facilitado la creación de los diagramas recogidos a lo largo de toda la memoria. Para la parte de la planificación se ha utilizado el programa GanttProject [11], que nos ha permitido construir el diagrama en el que se recogen todas las fases con sus fechas correspondientes. Por último, se ha utilizado Gimp [12], programa de edición de imágenes con el que se han editado todas las capturas de pantalla. Cabe destacar que todos los programas mencionados anteriormente son de software libre.

### **9.3 Futuro del proyecto**

Como se comentó durante la especificación de los requisitos funcionales, la aplicación aprovecha todas las funcionalidades ofrecidas por los 'Mobile Web Services' de Moodle. En el caso de que Moodle ampliará las posibilidades a través de esos servicios, tendríamos que introducir esos cambios en nuestra aplicación. Por ejemplo, en un futuro se incorporarán la opciones de participar en foros y realizar test, opciones ya disponibles en un sitio Moodle pero no a través de los 'Mobile Web Services'.

Alternativamente, y si fuera necesario, nosotros podríamos implementar cada una de esas funcionalidades y añadirlas como servicios en los diferentes sitios Moodle en los que queramos hacer su uso. El principal hándicap sería que en el caso de que el sitio no tenga incorporado ese servicio, la aplicación no tendría permitida esa posibilidad. Además, si realizamos algún cambio sobre ese servicio, tendríamos que cambiarlo en todos los sitios Moodle en los que se haya aplicado. Por tanto, inicialmente se desecha esta opción ya que perderíamos el alcance de nuestra aplicación (cualquier sitio Moodle de versión 2 o superior y que tenga habilitados los 'Mobile Web Services').

---

### ***Referencias y Bibliografía***

---

- [1] Web oficial de la plataforma Eclipse  
<http://www.eclipse.org/>
- [2] Lenguaje de programación Java  
[http://www.java.com/es/download/faq/whatis\\_java.xml](http://www.java.com/es/download/faq/whatis_java.xml)
- [3] Web oficial de Android  
<http://developer.Android.com>
- [4] Kit de desarrollo para Android  
<http://developer.Android.com/sdk/index.html>
- [5] Plugin ADT para Eclipse  
<http://developer.Android.com/tools/sdk/eclipse-adt.html>
- [6] Librería para convertir JSON en objetos Java  
<https://code.google.com/p/google-gson/>
- [7] Librería para el desarrollo de gráficas  
<http://Androidplot.com/>
- [8] Sistema de gestión de base de datos SQLite  
<http://www.sqlite.org/>
- [9] Documentación oficial de Moodle  
[http://docs.Moodle.org/all/es/P%C3%A1gina\\_Principal](http://docs.Moodle.org/all/es/P%C3%A1gina_Principal)
- [10] Página oficial sobre la herramienta BOUML  
<http://bouml.free.fr/index.html>
- [11] Página oficial sobre la herramienta GanttProject  
<http://www.ganttproject.biz/>

- [12] Página oficial de la herramienta de edición de imágenes Gimp  
<http://www.gimp.org/>
- [13] Foro de consulta para desarrolladores en general  
<http://stackoverflow.com/>
- [14] Foro de programación en Android  
<http://www.anddev.org/>
- [15] Wikipedia, la enciclopedia libre  
<http://www.wikipedia.org/>
- [16] Página principal de la forja de la red Iris  
<https://forja.rediris.es/>
- [17] Página oficial de la Free Software Foundation  
<https://www.fsf.org/>
- [18] Genymotion  
<http://www.genymotion.com/>
- [19] Pruebas de interfaz gráfica  
[http://developer.android.com/tools/testing/testing\\_ui.html](http://developer.android.com/tools/testing/testing_ui.html)
- [20] Stallman, Richard M. '*Software libre para una sociedad libre*', 4ª edición. GNU Press. Noviembre 2004. ISBN: 84-933555-1-0
- [21] Larman, Craig. '*Applying UML and Patterns*', 3ª edición. Prentice Hall, 2004. ISBN: 978-0131489066.
- [22] Pomés Olesti, Jaume. 'Introducción a Android y sus aplicaciones'. PFC de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla. Septiembre de 2009.
- [23] Gálvez Rojas, Sergio y Ortega Días, Lucas. '*Java a Tope*', Universidad de Málaga 2003. ISBN: 978-84-690-5677-6
- [24] Gramlich, Nicolas. 'Andbook' 2010. Libro online.
- [25] Murphy, Mark L. '*The Busy Coder's Guide to Advanced Android Development*', CommonsWare 2010. ISBN: 978-0-9816780-1-6
- [26] Allen, Grant & Owens, Mike. '*The definitive guide to SQLite*', 2ª edición. Apress 2010. ISBN: 978-1-4302-3225-4
- [27] Traducción al español de la Licencia Pública General GNU  
<http://www.viti.es/gnu/licenses/gpl.html>



# Apéndice A

---

## ***Publicación de la aplicación***

---

Si queremos distribuir la aplicación una vez completado el desarrollo de la misma, el SDK de Android compila todo el código Java y, junto con los ficheros de datos y recursos asociados (imágenes, etc.), crea un ‘paquete Android’: un archivo con extensión ‘**.apk**’ que representa a una única aplicación y que es el archivo utilizado por los dispositivos Android para instalar nuevas aplicaciones.

Estas aplicaciones, una vez instaladas, se ejecutarán como un nuevo proceso en el sistema Linux subyacente, utilizando su propio identificador único de usuario y su propia instancia de la máquina virtual Dalvik. Cuando la aplicación deja de utilizarse o cuando el sistema necesita memoria para otras tareas, el sistema operativo Android finaliza la ejecución del proceso, liberando los recursos. De esta forma, Android crea un entorno bastante seguro en el que cada aplicación, por defecto, sólo puede acceder a los componentes estrictamente necesarios para llevar a cabo su tarea, ya que cada proceso está aislado del resto y no puede acceder a partes del sistema para las cuales no tiene permisos.

La plataforma Android ofrece dos posibilidades para publicar una aplicación: dentro o fuera de **Google Play** (<http://play.google.com/>), que es la herramienta que Google pone a disposición de los desarrolladores para la publicación de sus aplicaciones, pero su uso no es obligatorio.

### ***A.1 Publicar fuera de Google Play***

De este modo, la aplicación puede darse a conocer de una manera sencilla y barata. Depende del propio desarrollador la distribución de la misma; esto incluye ponerla a disposición del usuario, publicitarla y adoptar un canal de comunicación con el usuario final, de manera que exista un intercambio de información sobre fallos, consejos, ayuda y actualizaciones, entre otras cosas.

Para llevar a cabo lo descrito anteriormente, es necesario compilar la aplicación y subir el archivo '.apk' resultante a alguna web para que pueda ser descargado. La promoción puede realizarse en redes sociales, foros, o mediante una página específica donde, además, se puede proporcionar información acerca de novedades sobre la aplicación.

La compilación debe realizarse en '*Modo Release*'. Esto se consigue pulsando con el botón derecho sobre la carpeta principal de nuestro proyecto en Eclipse y seleccionando '*Export Unsigned Application Package*' dentro del menú '*Android Tools*'. El archivo generado es que el debe ponerse a disposición del usuario para su descarga.

Sin embargo, para poder instalar la aplicación, el usuario debe configurar su terminal para aceptar aplicaciones que no hayan sido descargadas de Google Play, lo cual puede suponer un inconveniente.

Otro inconveniente de este método es que no podremos vender la aplicación de una manera tan sencilla como en Google Play, ni insertar publicidad como fuente de ingresos. Además, es más complicado obtener informes sobre el número de descargar o notificar al usuario final sobre posibles actualizaciones.

## **A.2      Publicar en Google Play**

La otra forma de publicar una aplicación es hacerlo en Google Play. Supone la forma más natural, útil y, a la larga, más fácil y beneficiosa. Este método, a pesar de tener un coste de 25\$, permitirá vender nuestra aplicación, insertarle publicidad, y tener acceso a una red de distribución mucho más extensa de lo que seguramente podríamos lograr mediante la autopromoción, dado que es una herramienta existente en todos los terminales Android.

El primer paso será darnos de alta como desarrolladores en Google Play accediendo a su web (<https://play.google.com/apps/publish/>) e introducir la información requerida, incluyendo los datos de pago. Una vez confirmados nuestros datos, accederemos a la consola de '*Google Play Developer*', configuraremos una cuenta de '*Google Wallet Merchant*' y ya podemos empezar a publicar aplicaciones.

Hay una pequeña diferencia con el caso anterior: en Google Play las aplicaciones deben ser firmadas, por lo que al crear el '.apk' habrá que elegir '*Export Signed Application Package*'. Cundo hayamos completado estos pasos, la aplicación ya puede ser publicada.

Para posteriores actualizaciones, es importante hacer dos cambios en el '*AndroidManifest.xml*' de la aplicación, de lo contrario Google Play no aceptará la subida. Los parámetros a modificar son los siguientes:

- `Android:versionCode = "1"`. Hay que incrementar una unidad cada vez que se realice una actualización.
- `Android:versionName = "1.0"`. Realmente no es obligatorio, pero se recomienda cambiar la numeración, de forma que sea orientativo para el usuario.

---

### ***Manual de usuario***

---

En este apartado detallaremos por completo el manual de usuario para poder utilizar todas las funcionalidades que nos presenta la aplicación una vez se encuentre instalada.

#### ***B.1 Pantalla de Login***

Será la primera pantalla que se mostrará al usuario cuando arranque la aplicación. El usuario podrá introducir la url del sitio Moodle al que se quiere conectar, su nombre de usuario y su password. Si el usuario aprecia que no existe su login debe de ponerse en contacto con el administrador del sitio Moodle al que desea conectarse para que efectúe su alta en el sistema.

En el caso de que se deje en blanco alguno de los campos o los datos introducidos no sean correctos, se mostrará al usuario un mensaje de error y se volverá de nuevo a la pantalla inicial.

Los botones disponibles son:

- **Login:** se comienza el proceso de validación con los datos introducidos por el usuario. Como se ha comentado anteriormente, en caso de producirse algún tipo de error, este será mostrado en pantalla.
- **Cuentas:** se muestra una lista de las cuentas con las que ya se ha hecho login anteriormente desde el terminal en el que se encuentra instalada la aplicación. El usuario podrá acceder con esas credenciales pulsando sobre alguna de las cuentas o eliminar una cuenta ya almacenada. Para ello deberá mantener pulsada varios segundos la cuenta que desea eliminar y se solicitará confirmación de la operación. En caso de producirse algún error en alguna de las operaciones descritas, se mostrará un mensaje de error. En la parte superior aparece un botón **Volver** para regresar a la pantalla de Login.

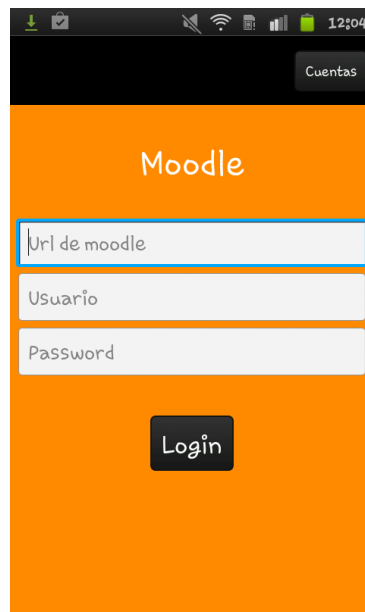


Figura B.1: Pantalla de Login

## B.2 Pantalla Principal

Una vez se han validado nuestras credenciales de acceso al sitio Moodle, se muestra la pantalla principal de la aplicación. En ella podemos observar 6 botones con las diferentes zonas que componen nuestra aplicación, según la funcionalidad con la que se encuentran relacionada. Los botones son:

- **Subir:** se accede al menú que muestra todas las alternativas para subir ficheros al espacio privado del usuario.
- **Participantes:** al pulsar sobre esta opción se mostrará primero una lista con los cursos a los que el usuario tiene acceso. Posteriormente podremos ver los participantes de cada uno de los cursos.
- **Cursos:** lista de cursos asignados al usuario y sus correspondientes contenidos (tanto como módulos como secciones).
- **Estadísticas:** se dibujan las gráficas con los datos estadísticos del usuario en cuestión para el terminal desde el que se encuentra accediendo.
- **Web:** muestra la versión web del sitio Moodle al que nos encontramos conectados.
- **Ayuda:** acceso a la web con la documentación oficial de Moodle.



Figura B.2: Pantalla Principal

### B.3 Subidas

Cuando accedemos a la zona de subidas se muestra una pantalla con las diferentes opciones que tenemos. Se trata de un nuevo menú en el que existen 3 botones, dependiendo de lo que queramos subir y cómo:

- **Buscar foto en álbum:** nos abre la galería de imágenes de nuestro terminal para que seleccionemos la foto que deseamos subir a nuestro espacio privado. Si no encontramos ninguna imagen que deseamos subir, podemos volver al menú de subidas.
- **Realizar foto o vídeo:** muestra la cámara del terminal. Según lo que deseamos, haremos una foto o grabaremos un vídeo y, si lo deseamos, podemos subirlo al espacio privado. En caso de que no nos convenza tenemos la opción de poder hacerlo o grabarlo de nuevo. En cualquier momento tenemos la opción de regresar al menú de subidas.
- **Grabar audio:** nos muestra una pantalla con los siguientes botones:
  - **Grabar / Parar:** al pulsar sobre este botón comienza a grabarse el fichero de sonido. Para detener la grabación, debemos pulsar de nuevo el mismo botón.
  - **Reproducir / Parar:** reproduce el fichero de audio que acabamos de grabar. Si queremos parar la reproducción, tendremos que pulsar de nuevo el botón.
  - **Enviar:** sube a nuestro espacio privado el fichero que acabamos de grabar.

- **Volver:** se retorna al menú de subidas.

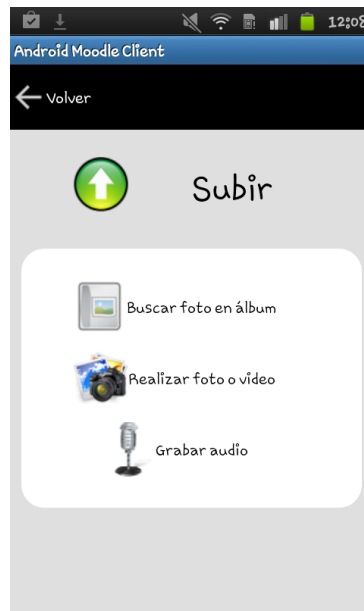


Figura B.3: Pantalla Subidas

## B.4 Participantes

Lo primero que se muestra al pulsar el botón de participantes en el menú principal es una lista con los cursos a los que el usuario tiene acceso. Llegados a este punto, tendrá que seleccionar el curso (pulsando sobre él) del que quiere conocer los participantes. De este modo, se muestra una lista con el nombre de todos los participantes que pertenecen al curso seleccionado. También tenemos la opción de regresar al menú principal pulsando sobre **Volver**.

Una vez tenemos delante la lista de participantes, podemos pulsar sobre uno de ellos si queremos acceder a información detallada del mismo. Además de la información detallada, aparecen los siguientes botones:

- **Enviar mensaje:** se muestra un cuadro en el que escribiremos el mensaje que queremos enviar al usuario seleccionado. Si finalmente no deseamos enviar el mensaje, podemos volver a la información detallada del usuario.
- **Añadir nota:** se añade una nota haciendo referencia al usuario. Igualmente, podemos volver sin agregar nada de información.
- **Añadir a contactos:** se agrega toda la información relativa al usuario en la libreta de direcciones del terminal.

- **Volver:** se regresa a la lista de participantes pertenecientes al curso que habíamos seleccionado previamente.



Figura B.4: Pantalla Información Participante

## B.5 Cursos

Al igual que en la zona de participantes, al acceder a los cursos se muestra una lista con los cursos a los que el usuario tiene acceso. Si queremos acceder al contenido del curso, pulsaremos sobre él y se nos mostrará una pantalla que contiene los módulos y secciones que lo componen. También podremos regresar al menú principal pulsando sobre **Volver**.

Dentro del detalle del curso se nos presentan las siguientes opciones:

- **Descargar contenido:** si pulsamos sobre un contenido junto el que aparece una flecha hacia abajo, se descargará dicho contenido al terminal.
- **Visualizar contenido:** podemos abrir dicho contenido si tiene una flecha hacia la derecha. El terminal debe estar dotado del visor correcto para abrir ese tipo de contenido. Por ejemplo, ficheros de tipo Word, pdf... o enlaces a páginas web. Adicionalmente, podemos abrir el contenido de un tipo carpeta, mostrándose todos los ficheros que contiene. Una vez dentro de la carpeta, podremos realizar las mismas acciones sobre los contenidos o volver al curso en el que nos encontramos.



- **Volver:** se regresa a la lista de cursos accesibles por el usuario.



Figura B.5: Pantalla Curso

## B.6 Datos estadísticos

Cuando desde el menú principal pulsamos sobre el botón de estadísticas, se muestra una pantalla en la que se dibuja una gráfica que contiene los datos relativos al usuario conectado para el terminal desde el que se encuentra realizando la conexión. Se trata de una gráfica en la que por fecha, suma el número de:

- **Logins:** sumatorio por día de todos los logins a ese sitio Moodle que ha realizado el usuario conectado desde el terminal por el que está accediendo.
- **Subidas:** sumatorio por día de todos los ficheros (foto, vídeo y audio) que el usuario conectado ha subido a su sitio personal desde el terminal por el que está accediendo.
- **Cursos:** sumatorio por día de todos los accesos a cursos que ha realizado el usuario conectado desde el terminal por el que está accediendo.

Además, se podrá volver al menú principal pulsando sobre el botón de Volver del terminal. Los botones que se muestran son:

- **+**: hace zoom, acerca la imagen
- **-**: quita zoom, aleja la imagen

- **1:1:** se vuelve al tamaño original de la imagen (de deshacen todos los zooms)

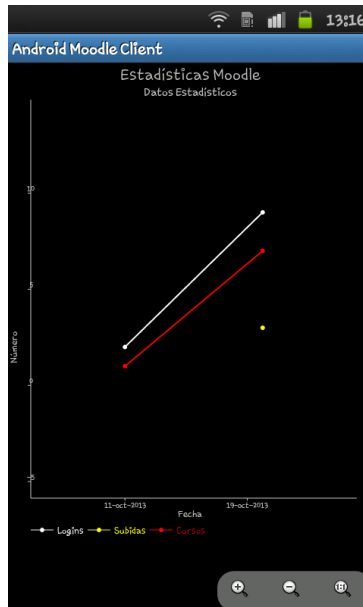


Figura B.6: Pantalla Estadísticas

## B.7 Versión Web

Otra de las opciones que tenemos desde la pantalla principal de la aplicación es pulsar el botón Web, que nos abrirá la versión web del sitio Moodle al que nos encontramos conectado. Dentro de la pantalla con la web aparecen los siguientes botones, propios de la navegación:

- **Atrás:** se regresa a la página anterior.
- **Siguiente:** se avanza a la página siguiente.
- **Recargar:** se vuelve a cargar el contenido de la página en la que nos encontramos.
- **Navegador:** se abre la versión web en el navegador del terminal en el que se encuentra abierta la aplicación.
- **Volver:** se regresa al menú principal.

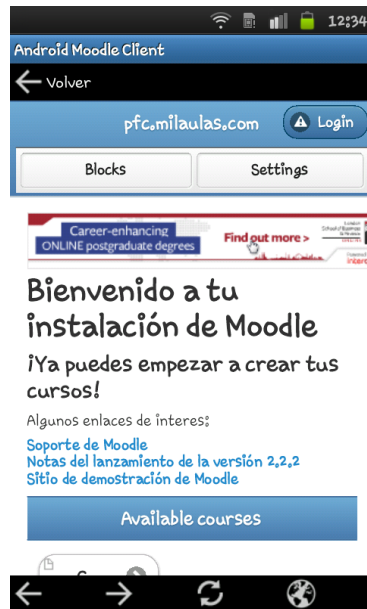


Figura B.7: Pantalla versión Web

## B.8 Ayuda

Finalmente, si pulsamos el botón de Ayuda, se cargará la web oficial con la documentación de ayuda de Moodle. Al igual que ocurría con la pantalla descrita en el punto anterior, aparecen los siguientes botones:

- **Atrás:** se regresa a la página anterior.
- **Siguiente:** se avanza a la página siguiente.
- **Recargar:** se vuelve a cargar el contenido de la página en la que nos encontramos.
- **Navegador:** se abre la versión web en el navegador del terminal en el que se encuentra abierta la aplicación.
- **Volver:** se regresa al menú principal.

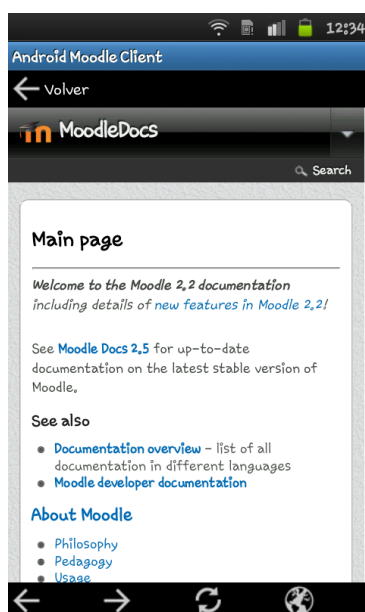


Figura B.8: Pantalla Ayuda

## Apéndice C

---

### ***Manual de instalación***

---

Se exponen a continuación todos los pasos necesarios para disponer de la aplicación en un terminal Android.

#### **C.1     *Prerrequisitos***

- Debemos disponer de un terminal con sistema operativo Android
- La versión del sistema operativo debe ser Android 2.1 o superior

**Nota:** recuerda que para poder conectarnos desde la aplicación a un sitio Moodle, éste debe ser versión 2.0 o superior. Además, deberá tener los 'Mobile Web Services' habilitados.

#### **C.2     *Instalación de la aplicación***

La instalación de la aplicación en nuestro terminal puede llevarse a cabo de 2 maneras:

##### ***a) Desde fichero apk***

El primer paso será descargar a nuestro PC el fichero con extensión .apk que contiene la aplicación. Una vez lo tengamos, tendremos que pasarlo a nuestro terminal mediante cable USB o a través de la tarjeta de memoria. Con ello, ya tendremos el fichero instalador en el terminal.

A continuación accederemos al menú 'Ajustes' del terminal y seleccionaremos la opción de 'Aplicaciones'. Ya una vez dentro, tenemos que activar la opción de 'Orígenes desconocidos' para permitir la instalación de aplicaciones no pertenecientes a Google Play.

Por último, nos ubicaremos en la carpeta donde previamente hemos guardado el fichero apk instalable de la aplicación y lo ejecutaremos. De este modo, la aplicación se encontrará instalada.



Figura C.1: Configuración para instalar apk

### ***b) Desde Google Play***

Accederemos al icono de Google Play desde nuestro terminal. Seguidamente, buscaremos la aplicación 'Cliente Moodle'. Una vez la hayamos encontrado, pulsaremos sobre '*Instalar*' para que se instale en nuestro dispositivo. Por último, tocaremos sobre '*Aceptar*' para aceptar los permisos requeridos por la aplicación y comenzar la descarga.

## **C.3 Puesta en funcionamiento**

Para arrancar la aplicación, simplemente tendremos que buscar el icono que se ha creado tras llevar a cabo la instalación y de este modo accederemos a la misma.

## C.4 Configuración del sitio Moodle

Para poder acceder desde nuestra aplicación a un sitio Moodle, éste debe ser versión 2.0 o superior y tener los ‘Mobile Web Services habilitados’. Esta última tarea deberá realizarla un usuario con privilegios de administrador. Para ello, activará la edición en la zona de Ajustes, accederá a Administración del Sitio y seleccionará Extensiones; dentro de esta rama buscará la pestaña de Servicios Web y accederá a Mobile. En la pantalla que se muestra deberá habilitar el servicio web para móviles. Las siguientes imágenes ilustran los pasos descritos:



Figura C.2: Configuración del sitio Moodle

## Apéndice D

---

### ***Ejemplos de la aplicación***

---

Con el objetivo de ejemplificar las diferentes situaciones en las que podemos hacer uso de la aplicación desarrollada, se describen en este apéndice un par de casos prácticos.

#### ***D.1 Excursión a la Sierra de Cádiz***

Supongamos una clase de primer curso de Zoología y Botánica del Grado en Ciencias Ambientales. El profesor, como actividad práctica, propone a sus alumnos una salida a la Sierra de Cádiz para identificar las distintas especies vistas en clase. Se evaluará que cada alumno consiga fotografiar al menos 3 de esas especies autóctonas.

Para ello, los alumnos tendrán que disponer de un terminal Android con la aplicación que hemos desarrollado instalada y con conexión de datos móviles. Para demostrar que han adquirido los conocimientos teóricos vistos en clase, tendrán que subir a su espacio las fotografías que realicen durante la clase práctica. En dichas fotos tendrán que distinguirse claramente las especies autóctonas.

Finalmente el profesor de la asignatura accederá al espacio de cada alumno para evaluar las fotografías realizadas. Podrá asegurarse de que fueron tomadas durante la práctica verificando la fecha y hora a la que fueron subidas al sitio Moodle.

#### ***D.2 Gymkana por el centro de Sevilla***

El profesor de la asignatura Iconografía del Grado en Historia del Arte ofrece a sus alumnos la posibilidad de realizar un juego por el centro de Sevilla. Dicho juego consistiría en una



gymkana en la que se visitarían distintos lugares emblemáticos de la ciudad y que han sido estudiados en clase.

Para poder participar, los alumnos tienen que disponer de un terminal Android con nuestra aplicación instalada y con conexión de datos móviles. La dinámica del juego será bastante divertida. El profesor colgará en su curso las diferentes pistas (correspondientes a cada prueba) en un archivo que los alumnos podrán descargar y visualizar en su terminal, pero que requiere una clave para poder ser abierto (se encuentra protegido). La manera de conseguir esa clave es escribiendo un mensaje privado al profesor a través de la aplicación; lo harán cada vez que lleguen al final de una prueba y el profesor, para dársela, le formulará una pregunta relacionada con la prueba que acaban de finalizar.

Con lo descrito anteriormente, los alumnos irán avanzando entre las distintas pruebas y el profesor podrá evaluarlos según el número de pruebas completado antes de que llegue el primer alumno a la meta.

# Apéndice E

---

## ***Licencia***

---

El texto del siguiente apéndice ha sido sacado íntegramente de [27].

Esta es una traducción no oficial de la Licencia Pública General GNU (GNU GPL) al español. No fue publicada por la Fundación para el Software Libre, y no establece legalmente los términos de distribución para software que utiliza la GNU GPL - solamente el texto original en inglés de la GNU GPL hace eso. De todas formas, esperamos que esta traducción ayude a los hispanohablantes a comprender mejor la GNU GPL.

### **LICENCIA PÚBLICA GENERAL GNU**

Versión 3, 29 de junio de 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Se permite la copia y distribución de copias literales de esta licencia, pero no está permitido modificarla.

### **Preámbulo**

La Licencia Pública General GNU (GNU GPL) es una licencia libre, sin derechos para software y otro tipo de trabajos.

Las licencias para la mayoría del software y otros trabajos prácticos están destinadas a suprimir la libertad de compartir y modificar esos trabajos. Por el contrario, la Licencia Pública General GNU persigue garantizar su libertad para compartir y modificar todas las versiones de un programa--y asegurar que permanecerá como software libre para todos sus usuarios. Nosotros, La Fundación de Software Libre, usamos la Licencia Pública General GNU para la mayoría de nuestro software; y también se aplica a cualquier trabajo realizado de la misma forma por sus autores. Usted también puede aplicarla a sus programas.

Cuando hablamos de software libre, nos referimos a libertad, no a precio. Nuestras Licencias Públicas Generales están destinadas a garantizar la libertad de distribuir copias de software

libre (y cobrar por ello si quiere), a recibir el código fuente o poder conseguirlo si así lo desea, a modificar el software o usar parte del mismo en nuevos programas libres, y a saber que puede hacer estas cosas.

Para proteger sus derechos, necesitamos evitar que otros le nieguen esos derechos o le pidan renunciar a ellos. Por lo tanto, usted tiene ciertas responsabilidades cuando distribuye copias del software, o si lo modifica: responsabilidades que persiguen respetar la libertad de otros.

Por ejemplo, si distribuye copias de tales programas, gratuitamente o no, debe transmitir a los destinatarios los mismos derechos que usted recibió. Debe asegurarse que ellos también reciban o puedan conseguir el código fuente. Y debe mostrarles estos términos y condiciones para que conozcan sus derechos.

Los desarrolladores que usen la GPL GNU protegen sus derechos de dos formas: (1) imponen derechos al software, y (2) le ofrecen esta Licencia para que legalmente lo copie, distribuya y/o modifique.

Para proteger a desarrolladores y autores, la GPL expone claramente que no existe garantía alguna para este software libre. Para beneficio de ambos, usuarios y autores, la GPL establece que las versiones modificadas deberán estar identificadas como tales, para que cualquier problema no sea atribuido por error a los autores de versiones anteriores.

Algunos dispositivos están diseñados para negar al usuario la instalación o la ejecución de versiones modificadas del software que usan internamente, aunque el fabricante sí pueda hacerlo. Esto es completamente incompatible con el objetivo de proteger la libertad de los usuarios para modificar el software. Este tipo de abuso sistemático ocurre con productos de uso personal, que es precisamente donde es menos aceptable. Por tanto, hemos diseñado esta versión de la GPL para prohibir estas prácticas en esos productos. Si apareciesen problemas similares en otros ámbitos, estaremos preparados para extender estas prestaciones a las próximas versiones de la GPL, tanto como sea necesario para proteger la libertad de los usuarios.

Por último, todo programa está constantemente amenazado por las patentes de software. Los estados no deberían permitir que las patentes restrinjan el desarrollo y el uso de software en ordenadores de uso general; pero en aquellos que lo hagan, esperamos evitar el especial peligro que suponen las patentes, que aplicadas a un programa libre puedan hacerlo propietario en la práctica. Para prevenir eso, la GPL establece que las patentes no pueden usarse para convertir un programa en no-libre.

A continuación se exponen los términos y condiciones para la copia, distribución y modificación.

## TÉRMINOS Y CONDICIONES

### 0. Definiciones.

En adelante "Esta Licencia" se refiere a la versión 3 de la Licencia Pública General GNU.

"Copyright" también significa "leyes similares al copyright" que son aplicables a otro tipo de trabajos, tales como las máscaras de semiconductores.

"El Programa" se refiere a cualquier trabajo con copyright al que se haya aplicado esta Licencia. Cada beneficiario es asimilable a "usted". "Beneficiarios" y "destinatarios" pueden ser personas físicas u organizaciones.

"Modificar" un trabajo significa copiar o adaptar todo o parte de un trabajo, exceptuando la copia exacta, de manera que se requiera permiso de copyright. El trabajo resultante se denomina "versión modificada" de un trabajo anterior o trabajo "basado en" el trabajo anterior.

Un "trabajo amparado" puede ser tanto el Programa no modificado como un trabajo basado en el Programa.

"Difundir" un trabajo significa hacer cualquier cosa con él, sin permiso, que le haga directa o indirectamente responsable de infringir leyes cubiertas por copyright, excepto la ejecución en un ordenador o la modificación de una copia privada. La difusión incluye la copia, distribución (con o sin modificaciones), distribución pública, y en algunos países también otras actividades.

"Distribuir" un trabajo implica cualquier tipo de difusión que permite a la otra parte hacer o recibir copias. La mera interacción con un usuario mediante una red de ordenadores, sin transferir copia alguna, no se considera "distribución".

Una interfaz de usuario interactiva muestra "Avisos Legales Apropriados" siempre y cuando incluya características visuales apropiadas y destacadas que (1) muestren un aviso de copyright apropiado, y (2) indiquen al usuario que no existe garantía alguna para el trabajo (exceptuando las garantías que se hayan podido establecer), que los beneficiarios deben distribuir el trabajo según se establece en la presente Licencia, y cómo se puede ver una copia de esta Licencia. Si la interfaz muestra una lista de opciones o comandos, tales como menús, un elemento destacado en dicha lista cumple estos criterios.

### 1. Código Fuente.

El "código fuente" de un trabajo es el formato preferido para realizar modificaciones sobre él. "Código objeto" se refiere a cualquier formato del trabajo que no sea código fuente.

Una "Interfaz Estándar" se refiere a una interfaz que sea o bien un estándar oficial definido por una institución de estándares reconocida, o bien, en el caso de interfaces específicos para una determinado lenguaje de programación, una cuyo uso esté generalizada entre los desarrolladores que trabajan con ese lenguaje.

Las "Bibliotecas de Sistema" de un trabajo ejecutable incluyen a cualquier elemento, que no sea el trabajo completo, que (a) esté incluida/o de la misma forma que un componente principal, pero que no forme parte de ese componente principal, y (b) sólo sirva para habilitar la utilización del trabajo a través de ese componente principal, o para implementar un Interfaz Estándar para el cual está disponible una implementación pública en código fuente. Un "Componente Principal", en este contexto, se refiere a un componente principal y esencial (núcleo, sistema de ventanas y similares) del sistema operativo particular (en su caso) sobre el cual funcione el ejecutable, o un compilador utilizado para generar el trabajo, o un intérprete del código objeto utilizado para ejecutarlo.

La "Fuente Correspondiente" de un trabajo en código objeto se refiere a todo código fuente necesario para generar, instalar, y (en el caso de trabajos ejecutables) ejecutar el código objeto y modificar el trabajo, incluyendo guiones que controlen esas actividades. Sin embargo, no se incluyen las Bibliotecas de Sistema del trabajo, o herramientas de propósito general o programas gratuitos habitualmente disponibles y usados sin ninguna modificación para realizar estas actividades pero que no forman parte del trabajo. Por ejemplo, la Fuente Correspondiente incluye los archivos de definición de interfaz asociados con archivos fuente del trabajo, y el código fuente de las bibliotecas compartidas o subprogramas enlazados dinámicamente que el programa requiere por diseño, como la comunicación de datos intrínseca o el control de flujo entre esos subprogramas y otras partes del trabajo.

La Fuente Correspondiente no incluye necesariamente aquello que los usuarios pueden regenerar automáticamente a partir de otras partes de la Fuente Correspondiente.

La Fuente Correspondiente de un trabajo en código fuente es ese mismo trabajo.

## **2. Permisos Básicos.**

Todos los derechos garantizados por esta Licencia se otorgan como copyright del Programa, y se proporcionan de manera irrevocable siempre y cuando se cumplan las condiciones establecidas. Esta Licencia afirma explícitamente su permiso ilimitado para ejecutar el Programa sin modificaciones. El resultado de la ejecución de un programa amparado está cubierto por esta Licencia sólo en el caso de que la salida, por su contenido, constituya un trabajo amparado. Esta Licencia reconoce sus derechos de uso razonable u otro equivalente, tal y como determina la ley de copyright.

Usted podrá realizar, ejecutar y difundir trabajos amparados que no distribuya, sin condición alguna, siempre y cuando no tenga otra licencia más restrictiva. Podrá distribuir trabajos

amparados a terceros con el mero objetivo de que ellos hagan modificaciones exclusivamente para usted, o para que le proporcionen ayuda para ejecutar esos trabajos, siempre que cumpla los términos de esta Licencia distribuyendo todo el material de cuyo copyright no posee el control. Aquellos que realicen o ejecuten los trabajos amparados para usted deben hacerlo exclusivamente en su nombre, bajo su dirección y control, con términos que les prohíban realizar copias de su material con copyright al margen de la relación con usted.

La distribución bajo otras circunstancias se permite únicamente bajo las condiciones establecidas más abajo. No está permitido sublicenciar; la cláusula 10 lo hace innecesario.

### **3. *Protección de Derechos Legales de los Usuarios frente a Leyes Anti-Burla.***

Ningún trabajo amparado debe considerarse parte de una medida tecnológica efectiva, a tenor de lo establecido en cualquier ley aplicable que cumpla las obligaciones expresas en el artículo 11 del tratado de copyright WIPO adoptado el 20 de diciembre de 1996, o leyes similares que prohíben o restringen la burla de tales medidas.

Cuando distribuya un trabajo amparado, renuncia a cualquier poder legal para prohibir la burla de medidas tecnológicas mientras tales burlas se realicen en ejercicio de derechos amparados por esta Licencia respecto al trabajo amparado; además, usted negará cualquier intención de limitar el uso o modificación del trabajo con el objetivo de imponer, al trabajo de los usuarios, sus derechos legales o de terceros para prohibir la burla de medidas tecnológicas.

### **4. *Distribución de copias literales.***

Usted podrá distribuir copias literales del código fuente del Programa tal y como lo ha recibido, por cualquier medio, siempre que publique de forma clara y llamativa en cada copia el correspondiente aviso de copyright ; mantenga intactos todos los avisos que establezcan que esta Licencia y cualquier término no-permisivo añadido y acorde con la cláusula 7 son aplicables al código; mantenga intactos todos los avisos de ausencia de garantía; y proporcione a todos los destinatarios una copia de esta Licencia junto con el Programa.

Usted podrá cobrar cualquier importe o no cobrar nada por cada copia que distribuya, y podrá ofrecer soporte o protección de garantía mediante un pago.

## **5. Distribución de Versiones Modificadas de Código.**

Usted podrá distribuir un trabajo basado en el Programa, o las modificaciones que lo producen a partir del Programa, como código fuente en virtud de los términos establecidos en la cláusula 4, siempre que cumpla todas las condiciones siguientes:

- a) El trabajo debe incluir avisos destacados indicando que usted lo ha modificado y dando una fecha pertinente.
- b) El trabajo debe incluir avisos destacados indicando que está realizado a tenor de lo dispuesto en la presente Licencia y en cualquier otra condición añadida en virtud de la cláusula 7. Este requisito modifica el requisito de "mantener intactos todos los avisos" expuesto en la cláusula 4.
- c) En virtud del presente documento, usted deberá aplicar la licencia al trabajo completo, como un todo, a cualquier persona que esté en posesión de una copia. Por lo tanto, esta Licencia se aplicará junto con cualquier otra condición adicional aplicable de la cláusula 7, al conjunto completo del trabajo y todas y cada una de sus partes, independientemente de como sean agrupadas o empaquetadas. Esta Licencia no permite ser aplicada al trabajo de ninguna otra forma, pero no se anula dicho permiso si usted lo ha recibido por separado.
- d) Si el trabajo tiene interfaces de usuario interactivos, cada uno debe mostrar Avisos Legales Apropriados; sin embargo, si el Programa tiene interfaces interactivos que no muestran Avisos Legales Apropriados, su trabajo no tiene porqué modificarlos para que lo hagan.

Un conjunto o recopilación formado por un trabajo amparado y otros trabajos distintos e independientes, que por su naturaleza no sean ampliaciones del trabajo amparado, que no se combinen con él de alguna forma para dar lugar a un programa mayor, y que estén ubicados en un medio de distribución o almacenamiento, se denomina "paquete" si la recopilación y su copyright al completo no son usados para limitar el acceso o los derechos legales de los usuarios de la recopilación, más allá de lo que permita el trabajo individual. La inclusión de un trabajo amparado en un paquete no hace aplicable esta Licencia al resto de elementos del paquete.

## **6. Distribución de código No-fuente.**

Usted podrá distribuir el código objeto de un trabajo amparado en virtud de los términos de las cláusulas 4 y 5, siempre que también distribuya las Fuentes Correspondientes en código máquina, de acuerdo con los términos establecidos en esta Licencia, de alguna de las siguientes maneras:

- a) Distribuir el código objeto en, o embebido en, un producto físico (incluyendo medios de distribución físicos), acompañado de las Fuentes Correspondientes en un medio físico duradero y que sea utilizado habitualmente para el intercambio de software.
- b) Distribuir el código objeto en, o embebido en, un producto físico (incluyendo medios de distribución físicos), acompañado de una oferta por escrito, válida al menos durante tres años y válida durante el tiempo en el que usted ofrezca recambios o soporte para ese modelo de producto, con el fin de ofrecer al poseedor del código objeto (1) una copia de las Fuentes Correspondientes a todo el software del producto que esté cubierto por esta Licencia, en un medio físico duradero habitual para el intercambio de software, a un precio no mayor que su coste razonable por distribuir físicamente las fuentes, o (2) acceso para copiar las fuentes correspondientes desde un servidor de red sin coste alguno.
- c) Distribuir copias individuales del código objeto junto con una copia de la oferta por escrito para/con el fin de proporcionar las Fuentes Correspondientes. Esta alternativa sólo está permitida ocasionalmente, pero no de forma comercial, y solamente si usted recibió el código objeto junto con una oferta parecida, de acuerdo con la subcláusula 6b.
- d) Distribuir el código objeto ofreciendo acceso desde un lugar determinado (gratuitamente o mediante pago), y ofrecer acceso equivalente a las Fuentes Correspondientes de la misma forma y en el mismo lugar sin cargo añadido. No es necesario exigir a los destinatarios que copien las Fuentes Correspondientes junto con el código objeto. Si el lugar para copiar el código objeto es un servidor de red, las Fuentes Correspondientes pueden estar en un servidor diferente (gestionado por usted o terceros) que ofrezca facilidades de copia equivalentes, siempre que mantenga instrucciones claras junto al código objeto especificando dónde encontrar las Fuentes Correspondientes. Independientemente de qué servidores alberguen las Fuentes Correspondientes, usted seguirá obligado a asegurar que estarán disponibles durante el tiempo necesario para cumplir estos requisitos.
- e) Distribuir el código mediante transferencias entre usuarios, siempre que informe a otros usuarios dónde se ofrecen el código objeto y las Fuentes Correspondientes de forma pública sin cargo alguno, tal y como se establece en la subcláusula 6d.

Una parte separable del código objeto, cuyo código fuente esté excluido de las Fuentes Correspondientes como Biblioteca de Sistema, no necesita ser incluida en la distribución del código objeto del trabajo.

Un "Producto de Usuario" es tanto (1) un "producto de consumo", que se refiere a cualquier propiedad personal tangible habitualmente utilizada para fines personales, familiares o domésticos, o (2) cualquier cosa diseñada o vendida para ser incorporada como extensión/expansión para otro producto. Para determinar si un producto es un producto de



consumo, los casos dudosos se resolverán favoreciendo el amparo. En el caso de un producto concreto recibido por un usuario particular, "de uso habitual" se refiere al uso típico o corriente de ese tipo de producto, independientemente de la situación del usuario particular o de la forma en que el usuario concreto utilice, o pretenda o se espere que pretenda utilizar, el producto. Un producto es un producto de consumo independientemente de si el producto tiene usos sustancialmente comerciales, industriales o distintos del consumo, a menos que tales usos representen la única forma posible de utilizar el producto.

Las "Instrucciones de Instalación" para un Producto de Usuario se refieren a cualquier método, procedimiento, clave de autorización, u otro tipo de información necesaria para instalar y ejecutar una versión modificada de un trabajo amparado en ese Producto de Usuario a partir de una versión modificada de las Fuentes Correspondientes. Las instrucciones deben ser suficientes para asegurar el funcionamiento continuo del código objeto modificado sin ningún tipo de condicionamiento o intromisión por el simple hecho de haber sido modificado.

Si, bajo las premisas de esta cláusula, usted distribuye el código objeto de un trabajo en, o con un Producto de Usuario o específicamente para ser usado en el mismo, y la distribución forma parte de una transacción donde los derechos de posesión y uso del Producto de Usuario se transfieren al destinatario a perpetuidad o durante un plazo fijo de tiempo (independientemente de las características de la transacción), las Fuentes Correspondientes distribuidas bajo estos supuestos deben acompañarse de las Instrucciones de Instalación. Sin embargo, estos requerimientos no se aplican si ni usted ni terceros tienen posibilidad de instalar el código objeto modificado en el Producto de Usuario (por ejemplo, el trabajo ha sido instalado en memoria de sólo lectura, ROM):

El requerimiento de proporcionar Información de Instalación no incluye el hecho de continuar proporcionando servicio de soporte, garantía, o actualizaciones para un trabajo que haya sido modificado o instalado por el destinatario, o para el Producto de Usuario en el que se haya modificado o instalado. El acceso a la red puede ser denegado cuando la propia modificación afecte materialmente y de forma adversa a la operación de la red o viole las reglas y protocolos de comunicación en la red.

Las Fuentes Correspondientes distribuidas, y las Instrucciones de Instalación proporcionadas de acuerdo con esta cláusula, deben figurar en un formato documentado públicamente (y con una implementación disponible para el público en código fuente), y no deben necesitar claves de acceso especiales para la descompresión, lectura o copia.

## **7. Condiciones adicionales.**

Los "Permisos Adicionales" son condicionantes que amplían los términos de esta Licencia permitiendo excepciones a una o más de sus condiciones. Los Permisos Adicionales que son aplicables al Programa completo deberán ser tratados como si estuviesen incluidos en esta

Licencia, hasta los límites de validez impuestos por las leyes aplicables. Si los permisos adicionales se aplicasen sólo a una parte del Programa, esa parte podría ser usada de forma independiente en virtud de dichos permisos, pero el Programa completo seguiría estando afectado por esta Licencia con independencia de los permisos adicionales.

Cuando distribuya una copia de un trabajo amparado, usted podrá opcionalmente eliminar cualquier permiso adicional de esa copia, o de alguna parte del mismo. (Los permisos adicionales pueden haber establecido que sea requerida su eliminación en ciertos supuestos si usted modifica el trabajo.) Usted puede establecer permisos adicionales en material añadido por usted a un trabajo amparado, sobre el cual tiene o podrá aportar sus permisos de copyright correspondientes.

Sin contravenir cualquier otra estipulación en esta Licencia, usted podrá, para el material que añada a un trabajo amparado, (si está autorizado por los poseedores de copyright de ese material) añadir condiciones a esta Licencia con los siguientes términos:

- a) Ausencia de garantía o limitación de responsabilidad diferente de los términos establecidos en las cláusulas 15 y 16 de esta Licencia; u
- b) Obligación de mantener determinados avisos legales razonables o atribuciones de autoría en el material o en los Avisos Legales Correspondientes mostrados por los trabajos que lo contengan; o
- c) Prohibir la tergiversación del origen del material, o solicitar que las diferencias respecto a la versión original sean señaladas de forma apropiada en las versiones modificadas del material; o
- d) Limitar la utilización de los nombres de los autores o beneficiarios del material con fines divulgativos; o
- e) Negarse a ofrecer derechos afectados por leyes de registro para el uso de marcas empresariales, registradas o de servicio; o
- f) Exigir indemnización a los autores y poseedores de la licencia de ese material, por parte de cualquier persona que distribuya el material (o versiones modificadas del mismo), estableciendo obligaciones contractuales de responsabilidad sobre el destinatario, para cualquier responsabilidad que estas obligaciones contractuales impongan directamente sobre los autores y poseedores de licencia.

Cualesquiera otras condiciones adicionales no-permisivas son consideradas "otras restricciones" en el contexto de la cláusula 10. Si el Programa, tal cual lo recibió, o cualquier parte del mismo, contiene un aviso indicando que está amparado por esta Licencia junto a una cláusula de restricción posterior específica, usted podrá suprimir esa cláusula. Si un documento de licencia contiene una restricción de este tipo pero permite modificar la licencia o la distribución en virtud de la presente Licencia, usted podrá añadirla al material de un trabajo

amparado por los términos de ese documento de licencia, siempre que dicha restricción no se mantenga tras la modificación de la licencia o la distribución.

Si añade condiciones para un trabajo amparado, a tenor de lo establecido en la presente cláusula, usted deberá ubicar, en los archivos fuente involucrados, una declaración de los términos adicionales aplicables a esos archivos, o un aviso indicando dónde localizar los términos aplicables.

Las condiciones adicionales, permisivas o no, deben aparecer por escrito como licencias separadas, o figurar como excepciones; de todas formas, los requisitos anteriores siempre son aplicables.

## **8. Cancelación.**

Usted no podrá distribuir o modificar un trabajo amparado salvo de la forma en la que se ha previsto expresamente en esta Licencia. Cualquier intento diferente de distribución o modificación será considerado nulo, y automáticamente cancelará sus derechos respecto a esta Licencia (incluyendo cualquier patente conseguida según el párrafo tercero de la cláusula 11).

Sin embargo, si deja de violar esta Licencia, entonces su licencia desde el poseedor del copyright correspondiente será restituida (a) provisionalmente, a menos que y hasta que el poseedor del copyright dé por terminada explícita y permanentemente su licencia, y (b) permanentemente, si el poseedor del copyright no le ha notificado por algún cauce de la violación no después de los 60 días posteriores al cese.

Además, su licencia desde el poseedor del copyright correspondiente será restituida permanentemente si el poseedor del copyright le notifica de la violación por algún cauce, es la primera vez que recibe la notificación de violación de esta Licencia (para cualquier trabajo) de ese poseedor de copyright, y usted subsana la violación antes de 30 días desde la recepción del aviso.

La cancelación de sus derechos según esta cláusula no da por canceladas las licencias de terceros que hayan recibido copias o derechos a través de usted con esta Licencia. Si sus derechos han finalizado y no han sido restituidos de forma permanente, usted no está capacitado para recibir nuevas licencias para el mismo material en virtud de la cláusula 10.

## **9. Aceptación no obligatoria por tenencia de copias.**

No está obligado a aceptar esta Licencia por recibir o ejecutar una copia del Programa. La distribución de un trabajo amparado surgida simplemente como consecuencia de la transmisión

entre usuarios para obtener una copia tampoco requiere aceptación. Sin embargo, únicamente esta Licencia le otorga permiso para distribuir o modificar cualquier trabajo amparado. Estas acciones infringen el copyright si usted no acepta las los términos y condiciones de esta Licencia. Por lo tanto, al modificar o distribuir un trabajo amparado, usted indica que acepta la Licencia.

### **10. Herencia automática de licencia para destinatarios.**

Cada vez que distribuya un trabajo amparado, el destinatario recibirá automáticamente una licencia desde los poseedores originales, para ejecutar, modificar y distribuir ese trabajo, al amparo de los términos de esta Licencia. Usted no será responsable de asegurar el cumplimiento por terceros de esta Licencia.

Una "transacción de entidad" es una transacción que transfiere el control de una organización, o todos los bienes sustanciales de una, o subdivide una organización, o fusiona organizaciones. Si la distribución de un trabajo amparado surge de una transacción de entidad, cada parte involucrada en esa transacción que reciba una copia del trabajo, también recibe todas y cada una de las licencias existentes del trabajo que la parte interesada tuviese o pudiese ofrecer según el párrafo anterior, además del derecho a tomar posesión de las Fuentes Correspondientes del trabajo a través de la parte interesada, si está en poder de dicha parte o se puede conseguir con un esfuerzo razonable.

Usted no podrá imponer restricciones posteriores en el ejercicio de los derechos otorgados o concedidos en virtud de la presente Licencia. Por ejemplo, usted no puede imponer a la licencia pagos, derechos u otros cargos por el ejercicio de los derechos otorgados según esta Licencia; además no podrá iniciar litigios (incluyendo demandas o contrademandas en pleitos) alegando que se infringen patentes por cambiar, usar, vender, ofrecer en venta o importar el Programa, o cualquier parte del mismo.

### **11. Patentes.**

Un "colaborador" es un poseedor de copyright que autoriza el uso del Programa o un trabajo en el que se base el Programa bajo los términos y condiciones establecidos en la presente Licencia. El trabajo con esta licencia se denomina "versión en colaboración" con el colaborador.

Todas las reivindicaciones de patentes en posesión o controladas por el colaborador se denominan "demandas de patente original", ya sean existentes o adquiridas con posterioridad, que hayan sido infringidas de alguna forma permitida por esta Licencia, al hacer, usar o vender la versión en colaboración, pero sin incluir demandas que sólo sean infracciones como consecuencia de modificaciones posteriores de la versión en colaboración. Para aclarar esta

definición, "control" incluye el derecho de conceder sublicencias de patente de forma que no contravenga los requisitos establecidos en la presente Licencia.

Cada colaborador le concede a usted una licencia de la patente no-exclusiva, global y libre de derechos bajo las reivindicaciones de patente de origen del colaborador, para el uso, modificación, venta, ofertas de venta, importación y otras formas de ejecución, modificación y redistribución del contenido de la versión en colaboración.

En los siguientes tres párrafos, una "licencia de patente" se refiere a cualquier acuerdo o compromiso expreso y manifiesto, cualquiera que sea su denominación, que no imponga una patente (como puede ser el permiso expreso para ejecutar una patente o acuerdos para no imponer demandas por infracción de patente). "Conceder" estas licencias de patente a un tercero significa llegar a tal tipo de acuerdo o compromiso que no imponga una patente al tercero.

Si usted distribuye un trabajo amparado, conociendo que está afectado por una licencia de patente, y no están disponibles de forma pública para su copia las Fuentes Correspondientes, sin cargo alguno y bajo los términos de esta Licencia, ya sea a través de un servidor de red público o mediante cualquier otro medio, entonces usted deberá o bien (1) permitir que sean públicas las Fuentes Correspondientes, o (2) tratar de eliminar los beneficios de la licencia de patente para este trabajo en particular, o (3) tratar de extender, de una forma que no contravenga los requisitos de esta Licencia, la licencia de patente a terceros. "Conocer que está afectado" significa que usted tiene conocimiento real de que, para la licencia de patente, la distribución del trabajo amparado en un determinado país, o el uso del trabajo amparado por sus destinatarios en un determinado país, infringiría una o más patentes existentes en ese país que usted considera aplicables por algún motivo.

Si, de conformidad con alguna transacción o acuerdo(o en un proceso relacionado con ellos), usted distribuye o distribuye con fines de distribución , un trabajo amparado, concediendo una licencia de patente para algún tercero que reciba el trabajo amparado, y autorizándole a usar, distribuir, modificar o distribuir una copia específica del trabajo amparado, entonces la licencia de patente que usted otorgue se extiende automáticamente a todos los receptores del trabajo amparado y cualquier trabajo basado en el mismo.

Una licencia de patente es "discriminatoria" si no incluye dentro de su ámbito de cobertura, prohíbe el ejercicio, o está condicionada a no ejercitar uno o más de los derechos que están específicamente otorgados por esta Licencia. Usted no debe distribuir un trabajo amparado si está implicado en un acuerdo con terceros que estén relacionados con el negocio de la distribución de software, en el que usted haga pagos relacionados con su actividad de distribución del trabajo, y donde se otorgue, a cualquier receptor del trabajo amparado, una licencia de patente discriminatoria (a) en relación con las copias del trabajo amparado distribuido por usted (o copias hechas a partir de éstas), o (b) directa o indirectamente relacionadas con productos específicos o paquetes que contengan el trabajo amparado, a

menos que usted forme parte del acuerdo, o que esa licencia de patente fuese otorgada antes del 28 de marzo de 2007.

Ninguna disposición de esta Licencia se considerará como excluyente o limitante de la aplicación de cualquier otra licencia o defensas legales contra la violación de las leyes de propiedad intelectual a que pudiera tener derecho bajo la ley de propiedad intelectual vigente.

#### ***12. No condicionamiento de la libertad de terceros.***

Si a usted le son impuestas condiciones que contravienen las estipuladas en la presente Licencia (ya sea por orden judicial, acuerdo u otros), no quedará eximido de cumplir las condiciones de esta Licencia. Si usted no puede distribuir un trabajo amparado cumpliendo simultáneamente sus obligaciones con esta Licencia y con cualquier otra pertinente, entonces no podrá distribuirlo de ninguna forma. Por ejemplo, si usted se compromete con términos que le obligan a obtener derechos por la distribución a terceros, la única forma de satisfacer ambos condicionantes y esta Licencia es abstenerse completamente de distribuir el Programa.

#### ***13. Uso conjunto con la Licencia Pública General Affero GNU.***

Sin contravenir las disposiciones de la presente Licencia, usted tendrá permiso para enlazar o combinar cualquier trabajo amparado con otro trabajo amparado por la versión 3 de la Licencia Pública General Affero GNU y formar un solo trabajo combinado, y distribuir el trabajo resultante. Los términos de esta Licencia seguirán siendo aplicables a la parte formada por el trabajo amparado, pero los condicionantes especiales de la Licencia Pública General Affero GNU, en su cláusula 13, relativos a la interacción mediante redes, serán aplicables a la combinación de ambas partes.

#### ***14. Versiones Revisadas de esta Licencia.***

La Fundación para el Software Libre podrá publicar revisiones y/o nuevas versiones de la Licencia Pública General GNU de vez en cuando. Esas versiones serán similares en espíritu a la versión actual, pero podrán diferir en algunos detalles para afrontar nuevos problemas o situaciones.

A cada versión se le da un número distintivo. Si el Programa especifica que le es aplicable cierto número de versión de la Licencia Pública General o "cualquier versión posterior", usted tendrá la posibilidad de adoptar los términos y condiciones de la versión indicada o de cualquier otra versión posterior publicada por la Fundación para el Software Libre. Si el Programa no especifica un número de versión de la Licencia Pública General, usted podrá elegir cualquier versión que haya sido publicada por la Fundación para el Software Libre.

Si el Programa especifica que un apoderado/representante puede decidir qué versiones de la Licencia Pública General pueden aplicarse en el futuro, la declaración pública de aceptación que el apoderado/representante haga de una versión le autoriza a usted con carácter permanente a elegir esa versión para el Programa.

Versiones posteriores de la licencia podrán otorgarle permisos adicionales o diferentes. Sin embargo, no podrán imponerse obligaciones adicionales a cualquier autor o poseedor de copyright como consecuencia de que usted adopte una versión posterior.

### **15. Ausencia de Garantía.**

EL PROGRAMA NO TIENE GARANTÍA ALGUNA, HASTA LOS LÍMITES PERMITIDOS POR LAS LEYES APLICABLES. SALVO CUANDO SE ESTABLEZCA LO CONTRARIO POR ESCRITO, EL POSEEDOR DEL COPYRIGHT Y/O TERCEROS PROPORCIONARÁN EL PROGRAMA "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS MERCANTILES Y DE APTITUD PARA UN PROPÓSITO DETERMINADO. USTED ASUMIRÁ CUALQUIER RIESGO RELATIVO A LA CALIDAD Y RENDIMIENTO DEL PROGRAMA. SI EL PROGRAMA FUESE DEFECTUOSO, USTED ASUMIRÁ CUALQUIER COSTE DE SERVICIO, REPARACIÓN O CORRECCIÓN.

### **16. Limitación de Responsabilidad.**

EN NINGÚN CASO, SALVO REQUERIMIENTO POR LEYES APLICABLES O MEDIANTE ACUERDO POR ESCRITO, PODRÁ UN POSEEDOR DE COPYRIGHT, O UN TERCERO QUE MODIFIQUE O DISTRIBUYA EL PROGRAMA SEGÚN LO INDICADO ANTERIORMENTE, HACERLE A USTED RESPONSABLE DE DAÑO ALGUNO, INCLUYENDO CUALQUIER DAÑO GENERAL, ESPECIAL, OCASIONAL O DERIVADO QUE SURJA DEL USO O LA INCAPACIDAD DE USO DEL PROGRAMA (INCLUYENDO PERO SIN LIMITARSE A LA PÉRDIDA DE DATOS O LA PRESENTACIÓN NO PRECISA DE LOS MISMOS O A PÉRDIDAS SUFRIDAS POR USTED O TERCEROS O AL FALLO DEL PROGRAMA AL INTERACTUAR CON OTROS PROGRAMAS), INCLUSO EN EL CASO DE QUE EL POSEEDOR O UN TERCERO HAYA SIDO ADVERTIDO DE LA POSIBILIDAD DE TALES DAÑOS.

**17. Interpretación de las cláusulas 15 y 16.**

Si la ausencia de garantía y la limitación de responsabilidad descrita anteriormente no tuviesen efecto legal a nivel local en todos sus términos, los juzgados aplicarán las leyes locales que más se aproximen a la exención de responsabilidad civil en lo relativo al Programa, a menos que la copia del Programa esté acompañada mediante pago de una garantía o compromiso de responsabilidad.